



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO DE UN SISTEMA DE MEDICION AUTOMATICA DE PECES :

T E S I S

PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A :
CARLOS AGUILAR VARELA

FACULTAD DE INGENIERIA



TESIS CON FALLA DE ORIGEN

MEXICO, D. F.

1994



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A mi familia: Paty, Alex, David, Esleie, Ivone, Yola, Alex y Rafael
y en especial Jesús y Lourdes,
por su constante amor y paciencia.

A mis amigos, Andrés, Kent, Mauricio, y los demás
por su presencia y apoyo.

A mis compañeros de trabajo, en gratitud
por lo que me han enseñado.

A todos los que participaron en el
proyecto: Gabriel, Juan Salvador, Mauro, Rubén.

Y a Yolanda y a Dios, por que sin su sonrisa
no podría existir.

TABLA DE ABREVIATURAS

| | |
|--------|--|
| APROX. | aproximadamente |
| BAUDS | Unidad de transmisión en caracteres por segundo |
| CISC | Juego de códigos de instrucciones completo |
| DCE | Equipo de comunicación de datos |
| DTE | Equipo terminal de datos |
| EIA | Electronic Industries Association |
| EOT | Fin de transmisión (04h) |
| E/S | Entrada Salida |
| FIG. | Figura |
| FOSC | Frecuencia de oscilación |
| GND | Tierra |
| Kb | Kilobytes (miles de bytes) |
| LSB | Bit menos significativo |
| mA | miliamperes |
| MAD | Direcciones de memoria de datos |
| MAP | Direcciones de memoria de programa |
| MSB | Bit más significativo |
| PC | Computadora Personal (Personal Computer) |
| PROG. | Programa |
| PULG | Pulgadas |
| RISC | Juego de códigos de instrucciones reducido |
| RST | Reinicio (reset) |
| SDS | Desarrollo de sistemas por software (Software Development Systems) |
| SEG. | Segundos |
| SET | Conjunto |
| STX | Inicia transmisión (02h) |
| SxPY | Pulso "y", del estado "x" (ver apéndice) |
| TERM | Terminal del dispositivo |
| TMD | Tiempo medio |
| VCD | Voltaje de corriente directa |
| VDD | Voltaje de alimentación |
| VMAX | Voltaje máximo |
| VMIN | Voltaje mínimo |
| V PROM | Voltaje promedio |

NOTA: LOS NOMBRES DE LOS REGISTROS DEL 8031, SE ENCUENTRAN EN EL APÉNDICE.

RESUMEN

Ha solicitud del Instituto de Ciencias del Mar y Limnología se desarrolló en el Centro de Instrumentos de la UNAM, un equipo de medición y almacenamiento de las dimensiones de animales marinos denominado Ictiómetro automático. Dicho equipo de medición debía satisfacer las siguientes características básicas:

- Medir y almacenar el ancho y largo de un pescado, cuya máxima medida sería de 40 cm.
- Tener una capacidad de almacenamiento de hasta 1000 muestras.
- Alcanzar una resolución de 0.5 mm.
- Soportar las condiciones ambientales de trabajo normales de humedad, temperatura y corrosión de un buque en altamar

Para satisfacer los objetivos planteados se definieron dos etapas en el proyecto: desarrollo de un sistema de instrumentación por software (resultando una arquitectura con la posibilidad de ser usado en otras aplicaciones) y la instrumentación por hardware la aplicación de medir pescados (adecuando las rutinas de adquisición, el sensor y la etapa de acondicionamiento de la variable real a medir).

El sistema mínimo de procesamiento digital se desarrolló en una tarjeta de dimensión 18x15 cm. y está basado en el microcontrolador MC80C31 (de la familia MCS'51 de Intel). Cuenta con las siguientes características: a) Una memoria de almacenamiento de programas para la interacción sistema-usuario de hasta 64 Kbits en memoria de tipo EPROM; b) Una memoria de almacenamiento de datos de 64 kb en memoria EEPROM, con el objeto de que los datos se conserven en forma no-volátil; c) Una pantalla de despliegue alfanumérico de cristal líquido AND 491, de 32 caracteres en 2 líneas; d) Un sistema de protección tipo "perro guardián", para prevenir de posibles fallas en el programa; e) Un puerto serie proporcionando la posibilidad de funcionar enlazado a una computadora a través de la norma RS-232;

f) Un puerto de digital E/S de 9 líneas. De estas, 2 líneas pueden ser señales de interrupción y 2 de contadores; y g) La operación en forma portátil mediante una batería de 6V.

Para cumplir la función específica de medir pescados se le agregaron los siguientes elementos al sistema mínimo: a) Un teclado para la entrada de comandos; b) Un dispositivo apuntador tipo "ratón" (MOUSE) como sensor, cuyo funcionamiento se basa en un codificador óptico de movimiento; y c) Una etapa de acondicionamiento de señales.

Como resultado tenemos un sistema de medición de peces con las siguientes capacidades:

- Una resolución teórica de 1/200 pulgadas (0.027 cm.).
- Un límite de longitud máxima de 50 cm.
- Un almacenamiento de hasta 4000 datos.
- Un funcionamiento básico con una batería electrolítica.
- Un el enlace con el puerto serie de una computadora tipo PC para el proceso de los datos.
- Un sistema lo suficientemente robusto para soportar las condiciones ambientales.

ÍNDICE

| | |
|--|----|
| RESUMEN | 3 |
| ÍNDICE | 5 |
| I. INTRODUCCIÓN | 8 |
| II. PLANTEAMIENTO INICIAL | 11 |
| 2.1. ¿Por que un instrumento de medición de peces? | 11 |
| 2.2. Introducción a los sistemas con microprocesador | 13 |
| 2.3. El instrumento de medición. | 15 |
| 2.4. Metodología de diseño. | 17 |
| III. MÉTODO DE MEDICIÓN. | 20 |
| 3.1. Formulación de alternativas | 20 |
| 3.1.1. Investigación de las distintas alternativas para hacer la medición. | 21 |
| 3.1.1.1. Equipos de trabajo mecánico. | 21 |
| 3.1.1.2. Equipos estáticos. | 24 |
| 3.1.2. Investigación de las características de los distintos sensores. | 25 |
| 3.1.3. Evaluación de factibilidad. | 27 |
| 3.1.4. Elección del método de medición. | 28 |
| 3.2. Descripción de los codificadores de movimiento. | 28 |
| 3.2.1. Clasificación por lectura del código. | 29 |
| 3.2.2. Clasificación por fenómeno físico empleado. | 30 |
| 3.2.3. Clasificación por tipo de codificación | 33 |
| 3.3. Características del dispositivo apuntador. | 35 |
| 3.4. Acondicionamiento y procesamiento de señales. | 36 |
| 3.5. Características del decodificador en el dispositivo apuntador. | 40 |
| IV. ARQUITECTURA DEL SISTEMA. | 41 |
| 4.1. Elección de la arquitectura del sistema. | 41 |
| 4.2. Descripción de la Arquitectura del equipo | 46 |
| 4.2.1. Descripción del sistema mínimo de procesamiento. | 46 |
| 4.2.1.1. Memoria | 47 |
| 4.2.1.2. Comunicaciones. | 51 |
| 4.2.1.3. Despliegue. | 56 |
| 4.2.1.4. Perro guardián | 60 |
| 4.2.2. Descripción de la tarjeta de aplicación | 63 |
| 4.2.2.1. Teclado de comandos. | 63 |
| 4.2.2.2. Interface con el sensor. | 65 |
| 4.2.2.3. Fuente de poder. | 70 |

| | |
|--|-----|
| V. DESCRIPCIÓN OPERATIVA DEL SISTEMA..... | 73 |
| 5.1. El Ictiómetro Automático..... | 73 |
| 5.1.1. Medición y almacenamiento de datos..... | 74 |
| 5.1.2. Revisión y edición de datos..... | 76 |
| 5.1.3. Transferencia de datos a una Computadora..... | 77 |
| 5.1.4. Menús de opciones..... | 77 |
| 5.1.5. Verificación del perro guardián..... | 79 |
| 5.1.6. Estados de ahorro de energía..... | 81 |
| 5.2. Procedimiento de adquisición y conversión de datos..... | 81 |
| 5.2.1. Conteo de Pulsos..... | 82 |
| 5.2.2. Estructura de almacenamiento..... | 85 |
| 5.2.3. Operación del ratón..... | 86 |
| 5.2.4. Almacenamiento de datos..... | 87 |
| 5.3. Procedimiento de comunicaciones..... | 89 |
| 5.4. Procedimiento de manipulación y edición de la información..... | 95 |
| 5.4.1. Verificación del estado de la memoria..... | 95 |
| 5.4.2. Borrado de memoria..... | 95 |
| 5.4.3. Borrado de archivo..... | 96 |
| 5.4.3.1. Determinación de archivo..... | 96 |
| 5.4.3.2. Determinación de dirección..... | 97 |
| 5.4.3.3. Borrar parte de datos..... | 98 |
| 5.4.4. Eliminación o edición de datos erróneos..... | 99 |
| 5.4.4.1. Incremento de archivo..... | 100 |
| 5.4.4.2. Incremento de número de dato..... | 100 |
| 5.4.4.3. Edición..... | 101 |
| 5.4.4.4. Medición..... | 101 |
| VI. PRESENTACIÓN AL USUARIO..... | 102 |
| 6.1. Diseño del programa comunicación residente en la PC..... | 102 |
| 6.1.1. Características del puerto serial en la PC..... | 102 |
| 6.1.2. Manejo de interrupciones..... | 103 |
| 6.1.3. Programa de Comunicación en la PC..... | 106 |
| 6.1.4. Optimización de la comunicación..... | 108 |
| 6.2. Diseño y construcción de chasis y de regla de medición..... | 109 |
| 6.2.1. Diseño del Chasis..... | 109 |
| 6.2.2. Módulo de control..... | 110 |
| 6.2.3. Diseño de la regla de medición..... | 111 |
| VII. CONCLUSIONES..... | 113 |
| 7.1. Conclusiones del Proyecto..... | 113 |
| 7.2. Recomendaciones..... | 115 |

| | |
|---|-----|
| VIII. BIBLIOGRAFÍA..... | 117 |
| IX. APÉNDICE..... | 121 |
| A. Descripción funcional de los dispositivos apuntadores..... | 121 |
| B. Descripción de la familia de microcontroladores MCS'51..... | 125 |
| 1. Registros Internos..... | 125 |
| 2. Descripción Operativa del 80C31..... | 127 |
| a) Estructura y operación de los puertos..... | 128 |
| b) Configuraciones de entradas/salidas de puertos..... | 129 |
| c) Acceso a la memoria externa..... | 131 |
| d) Contadores/relojes..... | 132 |
| e) Interface Serial..... | 133 |
| (1) Descripción de la operación del puerto serial incluyendo la operación de los distintos modos..... | 134 |
| (2) Registro de Control puerto serial..... | 134 |
| (3) Tasa de bauds..... | 135 |
| f) Interrupciones..... | 136 |
| (1) Estructura de nivel de prioridad..... | 138 |
| (2) Manejo de interrupciones..... | 139 |
| g) Señal de reinicio (Reset)..... | 140 |
| h) Reinicio en el encendido..... | 141 |
| i) Operación de los modos de ahorro de energía..... | 142 |
| (1) Paro de operaciones..... | 142 |
| (2) Reducción de energía..... | 143 |
| C. Lenguaje Ensamblador y Programación de la familia MCS'51..... | 144 |
| 1. Estructura de memoria..... | 144 |
| a) Memoria de Programas..... | 144 |
| b) Memoria de datos..... | 144 |
| 2. Registros de Control..... | 146 |
| 3. Modos de direccionamiento..... | 152 |
| 4. Juego de Instrucciones..... | 155 |
| D. Implantación Física..... | 160 |
| E. Rutinas básicas..... | 170 |
| 1. Rutina Inicial..... | 170 |
| 2. Inicialización de la pantalla de despliegue..... | 170 |
| 3. Atención al Teclado..... | 172 |
| 4. Manejadores de Pantalla..... | 174 |
| 5. Rutinas de espera..... | 182 |
| 6. Conversión binario - decimal..... | 183 |
| F. Listado de programas..... | 186 |

I. INTRODUCCIÓN

Los sistemas de procesamiento digital se han desarrollado buscando una forma sencilla y eficiente de resolver problemas. Las ventajas que les proporciona el uso del microprocesador (flexibilidad, facilidad de uso y costo reducido) han permitido que sean usados en diversos campos del quehacer humano. Los equipos que actualmente se desarrollan no están orientados a un problema específico, sino que son sistemas genéricos formados con los elementos básicos para su operación. Cuando se aplican a una tarea concreta le son añadidos los elementos que requieren, tanto a nivel físico como a nivel lógico, para que cumplan dicha función. De esta forma, como ventaja un mismo sistema puede aplicarse a distintas tareas, resultando más fáciles de usar y más baratos. En suma, un equipo de estas características busca que la implantación de la solución de un problema se haga a través de un programa, con el mínimo de cambios físicos (como puede ser el cambiar de sensor). A este tipo de equipos se les conoce como Sistemas de desarrollo por Software (SOFTWARE DEVELOPMENT SYSTEMS, SDS)[13].

La construcción de un sistema mínimo de procesamiento permite que su adaptación a diversas aplicaciones requiera el mínimo de cambios físicos. El sistema mínimo debe cubrir requerimientos como: medios de almacenamiento de datos y de programa, capacidades de comunicación, despliegue de información y fácil enlace a otros elementos.

El presente trabajo surge de dos necesidades básicas:

1. La construcción de un equipo capaz de medir y almacenar las dimensiones de un animal marino, en condiciones ambientales.
2. La creación de un sistema mínimo de procesamiento digital que permita el desarrollar proyectos de distinta índole.

El proyecto Ictiómetro automático del Centro de Instrumentos de la U.N.A.M., esta dirigido a resolver dichas necesidades. Para realizar este proyecto se llevaron a cabo las siguientes etapas:

- Diseño y construcción de un prototipo del sistema mínimo de procesamiento.
- Diseño del método de medición (sensor).
- Diseño y construcción de la electrónica (después del sensor) necesaria para realizar la medición.
- Desarrollo del programa de enlace entre la PC y el sistema.
- Diseño del chasis del equipo.

La organización de este trabajo está dividida en las siguientes partes:

Planteamiento inicial. En el primer capítulo se explican las motivaciones que fundamentaron la realización del proyecto, nociones acerca de los conceptos a tratar, la metodología que se siguió para su realización y el proceso en que fue llevado a cabo.

Método de medición. Este capítulo aborda la selección del procedimiento de medición, las características del sensor y su modo de enlace. Estas características son vitales en la concepción del programa.

Arquitectura del sistema. Esta parte se desarrolla un análisis detallado del sistema mínimo de procesamiento. Al final del capítulo se detallan las partes que se integraron al sistema para realizar la medición de los peces.

Descripción operativa del sistema. La forma en que el equipo funciona y las rutinas que soportan su operación como instrumento de medición son presentadas en este capítulo.

Presentación al usuario. En esta sección se detallan las características físicas del chasis para que el instrumento trabaje adecuadamente dentro de las condiciones ambientales esperadas. El programa de enlace con la computadora PC, para la transferencia de datos también se describe en este capítulo.

Conclusiones. En esta sección se realiza un análisis de los resultados obtenidos, y se presentan una serie de recomendaciones para trabajos futuros.

Apéndices. En esta última parte se describe, el funcionamiento de los dispositivos apuntadores; la organización física, funcional, y operativa de la familia de microcontroladores MCS'51; La implantación física del sistema, y la explicación de las rutinas básicas de soporte del equipo. Por último se muestra el código del programa del sistema.

II. PLANTEAMIENTO INICIAL

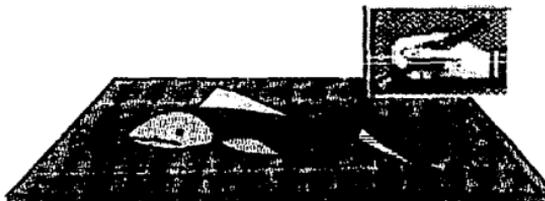
2.1. ¿Por que un instrumento de medición de peces?

Como parte de las actividades que realiza el Instituto de Ciencias del Mar y Limnología de la U.N.A.M. se encuentra el estudio de las características físicas de distintas especies marinas. Este estudio representa un arduo trabajo para el investigador, ya que deben ser tomadas una serie de mediciones bajo extremas condiciones ambientales en sesiones de trabajo de hasta 1000 muestras. Tradicionalmente estas mediciones se obtienen con regla y lápiz: sobre una tabla graduada, el pescado es medido y los datos son apuntados en una libreta (Fig. #1). Este proceso se repite para cada pescado. Sin embargo, debido sobre todo a la acumulación de errores humanos en la captura de los datos, los resultados de las medidas no tienen la precisión requerida y se toman pocas lecturas por el tiempo que se utiliza para realizar cada una. Además, se pierde más tiempo en transferir los datos a una computadora para su posterior procesamiento. De aquí surge la necesidad de buscar un método para automatizar, o por lo menos simplificar, las tareas de: medición, almacenamiento de datos y su posterior transferencia a una microcomputadora.

A petición del Instituto de Ciencias del Mar y Limnología, el Centro de Instrumentos de la UNAM desarrolló el proyecto número 1429 denominado Ictiómetro Automático. Este proyecto busca

resolver la problemática de medir, con el mínimo de error, la longitud y el ancho de una muestra de una población de distintos animales marinos; así como facilitar el manejo de los datos para su posterior procesamiento.

Fig. #1 Sistema usado para la medición



Las especificaciones, que a solicitud del Instituto de Ciencias del Mar y Limnología tenía que cubrir este instrumento son las siguientes:

1. La longitud máxima a medir es de 40 cm., tomada desde la punta de la cola del pescado a su otro extremo. Además, se tomara la medida de su ancho sin contar sus aletas .
2. El aparato requiere una resolución de por lo menos 0.5 mm.
3. Se espera tomar las medidas de una muestra de 1000 animales.
4. El sistema debe ser portátil.
5. Debe ser resistente a diversas condiciones de humedad y temperatura .
6. Costo reducido.
7. Fácil manejo y operación.

La creación de una arquitectura cerrada y específica a la solución del problema tiene la ventaja de que el equipo final estaría optimizado; es decir, que cumpliría la tarea aprovechando al máximo sus recursos. Al mismo tiempo tiene la desventaja de no poderse usar para otra cosa, por lo que el tiempo de diseño no tendría entonces una mayor utilidad; así mismo, su costo de fabricación sería elevado, ya que no tendría mayor demanda. Por esto se decidió crear un sistema mínimo de desarrollo por programa, que nos permite el desarrollar otro tipo de proyectos con pocas modificaciones físicas al sistema; por lo general basta cambiar el método de adquisición de los datos (sensor). De esta forma, la mayor parte de la implantación se realizaría en el programa de soporte del equipo. Esta filosofía en el diseño de sistemas se conoce como SDS (Software Development System)[referencia bibliográfica 13].

Nuestro primer objetivo es, por lo tanto, diseñar y construir un sistema de procesamiento y almacenamiento de datos, o SISTEMA MÍNIMO, capaz de:

- Permitir su reprogramación constantemente.
- Almacenar datos.
- Estar protegido contra errores.
- Poder comunicarse con el usuario.
- Poder comunicarse con otros sistemas.

Este sistema mínimo sería aplicado entonces como un instrumento capaz de realizar la medición y almacenar los resultados de manera no-volátil, para su posterior transferencia a una computadora.

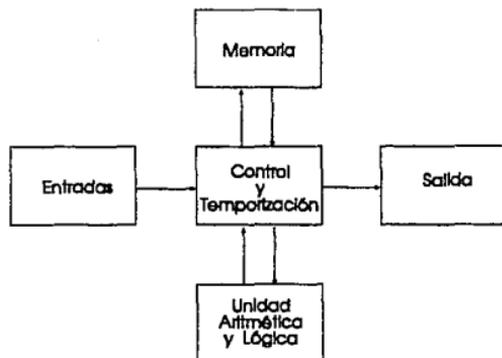
2.2. Introducción a los sistemas con microprocesador

El desarrollo de los semiconductores, su mayor producción y menor precio, a llevado a la aplicación de sistemas de procesamiento digital para la solución de problemas comunes. Los sistemas de procesamiento digital tienen ventajas sobre sistemas analógicos: Costo reducido, versatilidad, un más fácil manejo de la información. Pero tienen desventajas como su menor velocidad de procesamiento.

Los avances técnicos han permitido reducir las desventajas, además de que la fabricación de sistemas híbridos permite el aprovechamiento de lo mejor de ambas tecnologías.

Un sistema de procesamiento digital esta compuesto por lo elementos básicos [5] mostrados en la figura #2.

Fig. #2 Elementos básicos de un sistema de procesamiento digital



Estos elementos forman parte de cualquier sistema de procesamiento digital. Cuando se aplican a un trabajo específico, es necesario programar el equipo para que utilice los elementos de acuerdo a la función que va a realizar. El problema se reduce a emplear adecuadamente los recursos del sistema, o sea, a elaborar un algoritmo capaz de llegar a una solución correcta.

El microprocesador, inventado hace más de 20 años [17], hizo su propia revolución gracias a la integración de varios elementos genéricos en una sola unidad, haciéndolo más eficiente, económico y con mayor flexibilidad. Generalmente el microprocesador está formado por la unidad de Control y Temporización del Sistema y por la Unidad Aritmética y Lógica. Estos dos elementos se conocen como Unidad Central de Proceso (CPU) [5]. Debido a esto, al microprocesador a veces se le nombra de esta forma.

Usualmente, el microprocesador está formado por otros elementos además del CPU, por ejemplo: osciladores, manejadores de memoria, etc.; pero sigue dependiendo de otros dispositivos externos para su operación como puertos de entrada/salida, memoria o temporizadores. Cuando a un microprocesador se le agregan internamente estos elementos, se habla de que el dispositivo es un MICROCONTROLADOR. Por lo general, un microcontrolador es un dispositivo mucho más autónomo, ya que normalmente tan solo requiere de elementos de almacenamiento, (internamente cuentan con una capacidad reducida) e interfaces de entrada/salida para trabajar.

La evolución de los microprocesadores ha dirigido su mejoramiento en distintas áreas [18]:

Longitud de datos a procesar. Desde los primeros de 4 bits (nibble), se recorrió el camino en 8 (byte), 16, 32 y en la actualidad 64 bits. Esto representa

una mayor exactitud en las operaciones.

Capacidad de almacenamiento. La cantidad de memoria direccionable ha aumentado por la necesidad de procesar mayor cantidad de datos y por la complejidad que alcanzan los programas. De los 2 Kbytes que manejaba el 4004 (primer

microprocesador de propósito general) actualmente se tiene una capacidad de hasta varios Gigabytes, (Miles de millones de bytes).

Integración. Se han alcanzado niveles de integración (miniaturización) de hasta millones de transistores en un solo circuito (ULSI), permitiendo que más elementos estén contenidos en una misma unidad, como memoria "cache", procesamiento de punto flotante, etc.

Velocidad. Gracias a esta miniaturización y a la tecnología de semiconductores, actualmente se alcanzan velocidades de hasta 50 Mhz en el reloj del microprocesador, al reducir cada vez más las demoras y los tiempos de respuesta de los dispositivos y evitando los problemas que se generan como el de ruido en las señales, aumento de temperatura, etc.

Complejidad en las instrucciones. Se han seguido dos caminos: la tecnología CISC se basa en el uso de juegos de instrucciones complejas, de gran poder (es decir que con una sola instrucción se realizan complejos procesamientos de los datos); por lo mismo se ejecutan lentamente (varios ciclos de reloj). Por otro lado la tecnología RISC utiliza juegos de instrucciones muy simples y que se ejecutan en tiempos extremadamente cortos (un ciclo de reloj), lo cual permite el procesamiento más rápido de instrucciones.

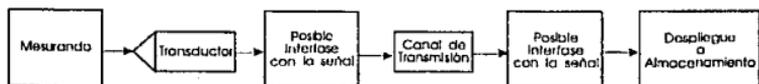
El microprocesador es el núcleo del sistema de procesamiento digital; de sus características depende el resto de la arquitectura del sistema. Esta arquitectura es la descripción de la organización entre el microprocesador y los dispositivos de apoyo externos, así como de la memoria [5]. Dicha arquitectura puede variar para darle mejores habilidades al sistema en ciertas áreas, como pueden ser la velocidad de acceso a la memoria o a los dispositivos de entrada/salida.

2.3. El instrumento de medición.

Un sistema de instrumentación está formado por los bloques mostrados en la figura #3 [2]. El transductor es el dispositivo capaz de

convertir las variaciones de una magnitud en variaciones de otra, aunque generalmente se le ha llamado transductor a un dispositivo capaz de convertir una magnitud no eléctrica en una señal eléctrica o viceversa [2]. El transductor es una de las partes más importantes del sistema, ya que de él depende la forma en que será tomada la medición. Todo transductor, para poder emplearse en un instrumento de medición, debe de tener caracterizada su respuesta.

Fig. #3 Diagrama de bloques de un sistema de instrumentación



Las principales características de un instrumento de medición son su rango dinámico, su resolución, la exactitud y su precisión [7].

El rango dinámico de un instrumento está dado por el menor y el mayor valor que puede medir, ya que más allá de estos valores, los resultados no son confiables.

La resolución nos indica cuál es la variación más pequeña que es capaz de detectar el instrumento y puede ser indicada como un porcentaje del rango, o como un valor absoluto en las unidades propias del instrumento.

La exactitud del instrumento es un valor de incertidumbre en nuestras lecturas; es decir, que tanto nuestra lectura puede estar desviadas de su valor real. Este parámetro es un reflejo de la confiabilidad del aparato. Generalmente es indicado como un porcentaje de la escala completa.

La precisión indica la capacidad de repetir la lectura. Es decir, que tanta diferencia existe entre las lecturas si medimos varias veces el mismo evento. Este parámetro nos indica la estabilidad del sistema y generalmente es expresado como un porcentaje. La exactitud de un sistema puede

mejorarse por calibración, pero nunca podrá ser menor que la precisión del sistema.

Para su estudio y de acuerdo a su aplicación, los instrumentos de medición se pueden clasificar, en instrumentos para [10]:

Monitoreo de procesos. Algunos procesos requieren de instrumentos de medición con fines de monitoreo, como son los termómetros o barómetros. Estos simplemente indican la situación de un ambiente y su respuesta no está dentro del sistema automático de control.

Control de procesos. Aquí los instrumentos de medición son usados como componentes de un sistema de control automático. En todos los esquemas de control con realimentación es necesario sensar una variable, por lo que se debe siempre de incorporar un elemento de medición.

Análisis experimental. Otro tipo de instrumentos de medición están orientados al trabajo experimental, con los que podemos recabar datos, evaluar nuestros resultados y complementar el trabajo teórico al compararlo con los resultados reales.

2.4. Metodología de diseño.

Para la elaboración del Ictiómetro automático, el proyecto se dividió en 5 etapas:

Determinación del método de medición. Se buscaron y evaluaron distintas alternativas para hacer la medición de las dimensiones de un pescado. Posteriormente, se establecieron las características del sensor y de la interface.

Diseño y construcción de un prototipo. Se diseñó el primer prototipo del sistema mínimo de procesamiento, con el objeto de evaluar los siguientes elementos:

- El funcionamiento del microcontrolador 8051 de Intel.
- El uso de una memoria de programas con capacidad de 32 Kbits.
- El uso de una memoria de datos en EEPROM, con capacidad de 64 Kbits.
- Una pantalla de LCD, de 32 caracteres.
- Una interface RS-232.

Conjuntamente, se realizó un primer diseño del protocolo de comunicación con la PC y de las rutinas básicas del sistema, como el acceso a puertos, a la memoria de datos y a la pantalla.

Prueba del sistema con el prototipo. En esta etapa se probaron las características esenciales del sistema. Se diseñó y construyó la interface entre el sensor y el sistema mínimo. Además se probaron los componentes de memoria, despliegue, e interface serial. Se diseñaron nuevas rutinas para probar las capacidades, entre ellas el uso de interrupciones, el conteo de distancia, el despliegue y almacenamiento de resultados.

Diseño y construcción del sistema de desarrollo. Fue diseñado agregando mejoras y corrigiendo fallas del primero. Este sistema fue realizado en 2 tarjetas: una contiene el sistema mínimo de procesamiento y la otra la interface con el sensor (tarjeta de aplicación). Como mejoras: le fue agregado el circuito de protección "perro guardián" (watch dog); se hizo uso de arreglos lógicos programables (PAL), en vez de dispositivos discretos, en la decodificación y en la interface con el sensor. También se implementó el uso de un teclado de 3 funciones y se añadió la fuente de poder en la tarjeta de aplicación. Se uso el microcontrolador 80C31 de Intel con capacidades de Hibernación. Además, se corrigieron errores en la organización de la tarjeta y en la decodificación de la memoria.

Conjuntamente se realizó el diseño y construcción del chasis y de la regla de medición que soporta al "ratón". Además, se implementó el programa de comunicación en la computadora PC que realiza el enlace y la recepción de información del Ictiómetro.

Prueba del sistema de desarrollo. En esta parte se integraron las distintas rutinas, por medio de un programa que accesa las funciones del

sistema. También se implementaron los estados de ahorro de energía, se mejoró el protocolo de comunicación con la PC y se trabajó en establecer un método de optimización de la transmisión. Por último se evaluaron las características finales del sistema.

III. MÉTODO DE MEDICIÓN.

INTRODUCCIÓN

El primer problema en el diseño de un sistema de instrumentación es determinar el método que se va a utilizar para realizar la medición.

Nuestros criterios de evaluación fueron de 2 tipos:

- Técnicos, como: resolución, rango, linealidad o respuesta al medio ambiente.
- No técnicos: costo, disponibilidad o tiempos de construcción.

En la primer sección: 1) **FORMULACIÓN DE ALTERNATIVAS**, se describen los posibles métodos en que pensamos para realizar la medición, las características de los sensores asociados a cada método. La formulación de alternativas y la evaluación que llevó a decidir el uso del dispositivo conocido como "ratón" para realizar la medición. En la sección 2) **DESCRIPCIÓN DE LOS CODIFICADORES DE MOVIMIENTO** se detallan las características más importantes de este tipo de sensores. su operación. En 3) **CARACTERÍSTICAS DEL DISPOSITIVO APUNTADOR**, se describe la conformación del dispositivo apuntador elegido; Se realiza un análisis de las señales del este dispositivo en 4) **ACONDICIONAMIENTO Y PROCESAMIENTO DE SEÑALES**, a partir de las cuales se desarrolló la interface con el sistema de procesamiento digital. Por último se describen las características específicas del sensor en la sección 5) **CARACTERÍSTICAS DEL DECODIFICADOR EN EL DISPOSITIVO APUNTADOR**.

3.1. Formulación de alternativas

Para la definición del método de medición fue necesario realizar las siguientes actividades: 1) Investigación de las distintas alternativas para hacer la medición; 2) Investigación de las características de los distintos sensores; 3) Evaluación de factibilidad; 4) Elección del método de medición.

3.1.1. Investigación de las distintas alternativas para hacer la medición.

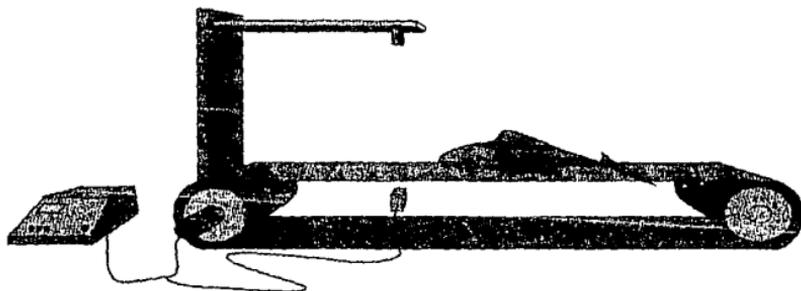
La primera tarea fue definir el método apropiado para hacer la medición de longitud. La diversidad de sensores capaces de medir una distancia nos llevo a definir, antes de escoger el sensor, cual sería la mejor forma para realizar la medición. Dependiendo del tipo del principio que se utilizaría para efectuar la medición se escogería al sensor. Los métodos con los que se podría realizar la medición los agrupamos según sus forma de operación en:

3.1.1.1. Equipos de trabajo mecánico.

Los definimos de esta forma por que funcionan con la aplicación de un trabajo mecánico que al mover, ya sea al sensor o al objeto a medir, proporcionan el estímulo necesario para hacer la lectura. Los sensores asociados a estos métodos detectan el movimiento, no la distancia. Es necesario entonces detectar el inicio y el final del objeto y a partir de la magnitud o la duración del estímulo se determina la distancia recorrida. Este trabajo puede ser realizado por medio de bandas sin fin, tornillos de potencia, etc.

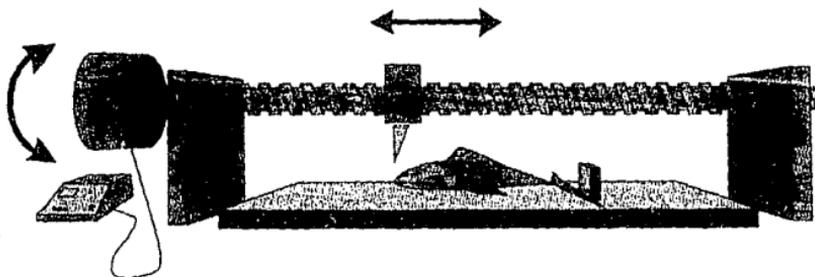
Banda sin fin. El espécimen se mueve sobre una banda sin fin y un dispositivo óptico sirve para iniciar y terminar la medición (fig. #4). También pueden ser usados sensores de movimiento para realizar la medición, como transductores de reluctancia, potenciométricos o codificadores ópticos. Este sistema puede ser completamente automático, de gran velocidad y con una buena resolución; sin embargo, sería de gran tamaño, por lo tanto difícil de construir y transportar.

Fig. #4 Propuesta de medición con banda sin fin



Tornillo de potencia. En este sistema, mostrado en la fig. #5, el movimiento lineal de una guía se transforma en un movimiento circular. Esto se logra con el uso de un tornillo de potencia; que se caracteriza por la eficiencia para mover su tuerca, eliminando la fricción con las paredes por medio del uso de balines. De esta forma el movimiento lineal de la tuerca es transformado en un giro del eje. Este movimiento puede ser fácilmente cuantificado añadiéndose un sensor de movimiento angular (potenciométrico, codificador óptico, etc.), por lo que la distancia sería un factor del número de vueltas que dio el eje. La resolución depende de la proporción entre el ángulo que gira y la distancia que recorren. El sistema

Fig. #5 Propuesta de medición con tornillo de potencia



tiene el problema de ser de un alto costo de construcción y mantenimiento por la pieza necesaria para alcanzar la precisión requerida.

Compás. En este método dos brazos, unidos en un extremo en forma de compás, se abren para cubrir la longitud del pez; como lo muestra la fig. #6. Se determina el ángulo y conociendo el tamaño de los brazos se puede calcular la longitud. Es necesario contar con un elemento mecánico para que este movimiento pueda ser amplificado y obtener mejores resultados. Sin embargo, este es su mayor problema, ya que sólo una fracción del arco completo será usado, por lo que el sensor debe de ser muy sensible ya que esta obligado a detectar hasta décimas de grado para lograr la resolución requerida.

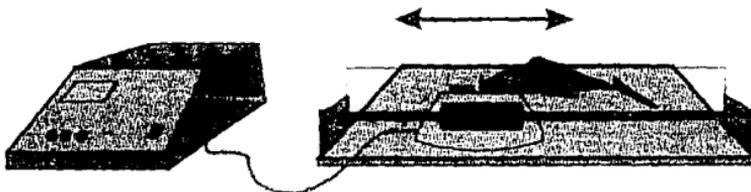
Fig. #6 Propuesta de medición con compás



Dispositivo apuntador. Los dispositivos apuntadores, denominados "ratón", son usados comúnmente en los sistemas de cómputo como medio de entrada de comandos. Utilizan distintos métodos para detectar y cuantificar el movimiento. En nuestro sistema, el ratón recorre longitudinalmente al pez, mandando señales del sentido y la magnitud de su desplazamiento. Dependiendo de su fabricante estos dispositivos

apuntadores entregan la información codificada de distintas formas; como consecuencia nos restringe su utilización a un tipo de "ratón".

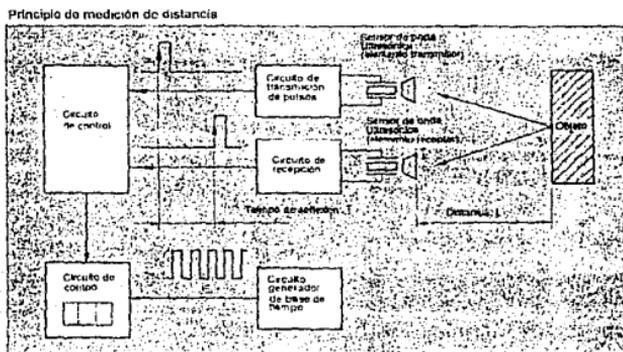
Fig. #4 Propuesta con dispositivo apuntador



3.1.1.2. Equipos estáticos.

En este tipo de dispositivos no se aplica ningún tipo de desplazamiento mecánico, ya que utilizan algún tipo de fenómeno físico propagativo, como puede ser luz o sonido.

Fig.#7 Operación genérica del radar



Sonares. Transmitimos una señal ultrasónica la cual es reflejada por algún objeto; midiendo el tiempo que tarda en regresar dicha señal se

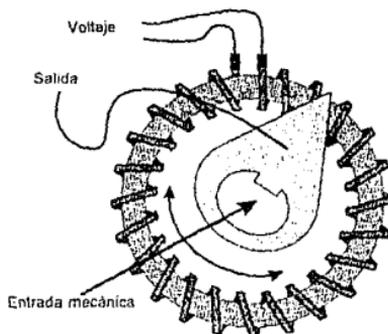
puede determinar la distancia a la que se encuentra el objeto (pues se conoce la velocidad del sonido). Estos sistemas son muy exactos, pero tienen la limitación de que solo pueden medir distancia mayores a la longitud de onda de la señal y que la exactitud de la medición se ve afectada por la frecuencia [20], por lo que el sensor debe de ser de alta frecuencia. La precisión de la medición se puede ver afectada por condiciones externas (como puede ser la temperatura, la presión atmosférica y la humedad), que provocan la variación de la velocidad del sonido.

3.1.2. Investigación de las características de los distintos sensores.

Existen diversas clases de sensores de desplazamiento y de posición capaces de cumplir con nuestro objetivo medir una longitud. Estos difieren de acuerdo al tipo de estímulo que requieren para su funcionamiento. En general, unos requieren de un trabajo mecánico, mientras otros funcionan con la detección de un objeto. A partir de los 5 posibles métodos descritos anteriormente, tres tipos de sensores fueron los que más nos interesaron para la realización del proyecto:

Transductores potenciométricos [10]. En este tipo de sensores un contacto deslizante (denominado "escobilla") recorre un elemento resistivo. La escobilla se mueve al estar conectada con la fuente del movimiento que va a medir. La conexión puede ser directa, si la escobilla esta montada directamente sobre el elemento que se mueve; o indirecta, por un medio mecánico como bandas o engranes. La salida es una fracción del voltaje aplicado en los extremos del potenciómetro y, dependiendo de la forma del acoplamiento con el

Fig.#8 Transductor resistivo



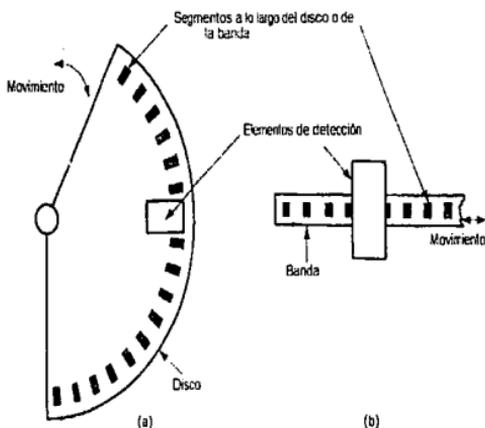
elemento móvil, será proporcional al movimiento realizado.

Algunos de estos tipos de potenciómetros están hechos de hilos delgados (de cerca de 0.01 mm. de diámetro) trenzados en superficies rectas o curvas. Pueden ser hechos de elementos como plástico conductor, película de carbón, película metálica o mezclas metal-cerámicas; mientras que las superficies sobre las que se enrollan son de material aislante. Las escobillas tienen distintas formas y son fabricadas con aleaciones de materiales preciosos o aleaciones de cobre. Pueden tener muelles tensores o ser de doble escobillas, con el fin de evitar los efectos de los golpes o de las vibraciones y se calibran a una presión de 4 a 15 gr. sobre el elemento resistivo. Son muy susceptibles a la contaminación y a la corrosión, por lo que deben estar bien sellados.

Al ser dispositivos analógicos requieren, para su uso en sistemas digitales, de algún tipo de conversión analógica-digital.

Codificadores Lineales y Angulares [10]. Estos dispositivos convierten el movimiento en una salida codificada digitalmente; de esta forma, desplazamientos angulares y lineales pueden ser directamente sentidos por un sistema digital, sin la necesidad de usar un convertidor analógico digital.

Fig. #9 Funcionamiento del codificador óptico



En estos dispositivos se detecta la presencia o ausencia de una señal de estímulo. Dicha señal indica la posición del instrumento, ya que el sensor o la fuente se mueven con el objeto. Dependiendo de la naturaleza de la señal que se usa como estímulo existen diferentes tipos de codificadores: ópticos, magnéticos y eléctricos.

Sensores ultrasónicos [10]. Este tipo de sensores son de un costo alto, por la electrónica que requieren ya que deben de ser inmunes a ruido de otras fuentes. Están generalmente contruidos en forma par (transmisor y receptor) con cerámicas piezoeléctricas. Estos materiales sufren una distorsión mecánica cuando se les aplica un voltaje (como es el caso del transmisor); en forma análoga, en el receptor se produce una diferencia de potencial cuando es sometido a algún tipo de deformación.

3.1.3. Evaluación de factibilidad.

En esta sección se describe el resultado de la evaluación de las alternativas a través de 2 tablas, una para los sensores y otra para las posibles formas de medición. Para seleccionar el método adecuado de medición de la longitud, tomamos en cuenta los siguientes puntos:

- Resolución y rango del sensor
- Exactitud y precisión del sistema
- Resistencia al medio
- Costo
- Complejidad de construcción
- Facilidad de uso

| CARACTERÍSTICAS DE LOS POSIBLES SENSORES | | | | |
|--|--------------------------------|---|---|-------|
| SENSOR | ALTERACIONES | RESOLUCIÓN | FUENTE DE ERROR | COSTO |
| Codificador Óptico | Temperatura | $1/2^6$ a $1/2^{22}$ | - Variación de la sensibilidad de los sensores. - Regularidad de las perforaciones | Medio |
| Transductores Potenciométricos | Temperatura | $1/2^6$ a ∞ | Relación vueltas distancia | Bajo |
| Sensor ultrasónico | Humedad, Temperatura y Presión | Es función de la exactitud de la medición del tiempo. | Variación de la velocidad del sonido | Alto |

| CARACTERÍSTICAS DE LOS POSIBLES MÉTODOS DE MEDICIÓN | | | | |
|---|------------------------------|------------------------|---------------------------------------|---------------------|
| MÉTODO DE MEDICIÓN | FACILIDAD DE USO | PORTABILIDAD | COMPLEJIDAD DE CONSTRUCCIÓN | COSTO |
| Banda sin fin | Muy simple al ser automático | Difícil de Transportar | Difícil Mecánica Fácil Electrónica | Muy Alto (Mecánica) |
| Tornillo de potencia | Simple al operar manualmente | Difícil de Transportar | Difícil Mecánica Fácil Electrónica | Muy Alto (Mecánica) |
| Compás | Simple al operar manualmente | Fácil de Transportar | Difícil Mecánica Fácil Electrónica | Reducido |
| Dispositivo apuntador | Simple al operar manualmente | Fácil de Transportar | Fácil Mecánica Fácil Electrónica | Reducido |
| Sonar | Muy simple al ser automático | Muy Fácil Transportar | Fácil Mecánica Fácil Electrónica | Medio (Electrónica) |

3.1.4. Elección del método de medición.

Se eligió el uso del dispositivo apuntador por ofrecer, a un costo reducido, una estructura completa; ya que se tienen los sensores y parte de la interface integrada en un sólo módulo. Cuenta con una resolución de 1/200 de pulgada (.0127 cm.), que es adecuada a las necesidades del equipo a diseñar.

3.2. Descripción de los codificadores de movimiento.

Como ya se ha mencionado, los dispositivos apuntadores generalmente funcionan con codificadores de movimiento o con métodos ópticos. Los primeros son más comunes por su menor costo y complejidad, además de que no requieren de una superficie especial para ser

accionados. Los codificadores de movimiento son transductores mecánicos a eléctricos, que traducen un desplazamiento en una o varias señales digitales [2]. Estas señales indican las características del movimiento (magnitud y sentido) derivadas de la lectura de un patrón codificado en un disco rotatorio o una escala móvil. Son usados comúnmente para detectar y cuantificar un movimiento. Este tipo de transductores son cada vez más usados, debido sobre todo a la importancia que han adquirido los sistemas de control numérico digital.

Los codificadores pueden clasificarse por las siguientes formas [2] :

- Método usado para la lectura del código: contacto o no-contacto.
- Fenómeno físico empleado para producir la salida: magnético, óptico, capacitivo y eléctrico.
- Tipo de codificación: series de pulsos incrementales (fig. #11) o palabra digital absoluta (fig. #10).

Fig. #10 Codificador de palabra absoluta



Fig. #11 Cod. de pulso incremental



A continuación se describen brevemente estas clasificaciones.

3.2.1. Clasificación por lectura del código.

Los codificadores de contacto se caracterizan por emplear el fenómeno de conducción eléctrica para realizar la codificación. Estas usan de una escobilla o una terminal sensora para hacer contacto con el código del

disco, que está formado por una serie de vías; las cuales son hechas de material conductor y están unidas en sus bases, por lo que al tocar las escobillas generan la señal de salida.

Cualquier patrón puede ser reproducido fotográficamente por un disco codificado. Estos pueden incluir vías uniformes o no uniformes, defasadas con respecto a otras, o hasta configuraciones inusuales para propósitos especiales. De cualquier forma, los codificadores más comunes son aquellos que tienen posiciones fijas, donde cualquier falta de uniformidad es fuente de errores. Esta falta de uniformidad es efecto del espaciamiento de los segmentos y la excentricidad de la vía. Un espaciamiento no uniforme de los segmentos produce un error de posición, mientras que la excentricidad de la vía produce un error que es una función senoidal del ángulo de la flecha.

Para realizar la lectura del código, los codificadores de no-contacto funcionan usando un fenómeno físico distinto al de conducción eléctrica. Los fenómenos físicos más comúnmente empleados son: magnéticos, capacitivos y ópticos. Estos se explican en la siguiente parte.

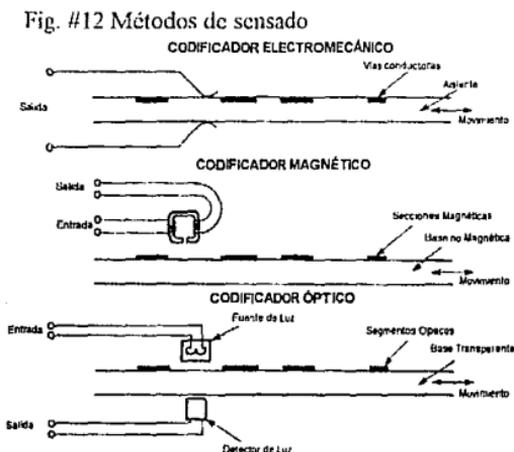
3.2.2. Clasificación por fenómeno físico empleado.

Codificadores magnéticos. Funcionan a partir de 3 métodos distintos: frecuencia de resonancia, saturación magnética del inductor o por cambio de magnetización. Independientemente del método, el cambio de estado se realiza por la acción del flujo magnético producido en el disco codificado. Este tipo de codificadores tienen una exactitud similar a la de los codificadores de contacto, ya que su resolución esta limitada por el tamaño de los puntos magnéticos y por la interacción entre ellos. Además, son susceptibles a la de radiación y los campos electromagnéticos; por lo que deben de ser aislados. Sin embargo tienen la ventaja de resistir condiciones ambientales adversas y poder funcionar a mayores velocidades, con una mayor duración al no existir un contacto directo entre el sensor y el disco.

Codificadores capacitivos. Este tipo de codificadores es el menos usado, aplicándose solamente a sistemas especiales. La lectura se realiza

midiendo la fase de una señal o por técnicas de control de frecuencia. Presentan muchos problemas prácticos en su diseño, manufactura y operación, por lo que se han realizado pocos sistemas de este tipo.

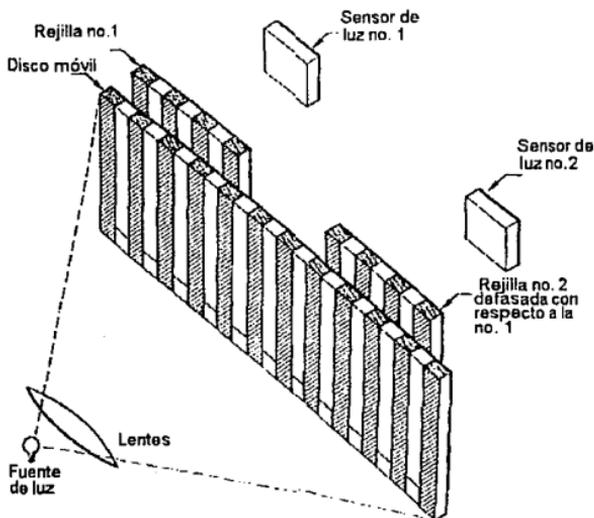
Codificador óptico. Este tipo de codificador fue el primero de los dispositivos de no contacto desarrollado para eliminar los problemas de los codificadores de contacto. Actualmente son de alta precisión y con gran eficiencia a altas velocidades. Pueden ser circulares o lineales, dependiendo del movimiento que se va a cuantificar, por lo que se aplican los mismos principios a ambos (ver fig. #12).



Los codificadores ópticos (tanto lineales, como circulares) están formados por segmentos opacos y transparentes, hechos generalmente de emulsión fotográfica, o de metal cuando se necesita gran durabilidad. La lectura es realizada por sensores fotoeléctricos colocados enfrente del codificador (fig. #13). La fuente de luz es provista por lamparas incandescentes o por LED's. Cuando el codificador se mueve, los segmentos opacos cortan el haz de luz entre los sensores y la fuente, modulando la salida de los sensores de acuerdo al código elegido. Algunos

elementos ópticos pueden ser añadidos para enfocar la luz en el disco codificado; como pueden ser espejos, lentes, fibra óptica, etc.

Fig. #13 Operación de los codificadores ópticos.



Para detectar la luz pueden ser usados diferentes dispositivos. Cada tipo de sensor responde en forma distinta a la presencia de luz. Las celdas fotovoltaicas generan una corriente eléctrica cuando son expuestas a la luz. En las fotoresistencias su capacidad de conducción eléctrica varía al ser estimuladas con luz. Los fotodiodos son similares a las fotoresistencias, con la diferencia de que con pequeñas áreas permiten una buena respuesta a alta frecuencia. Los fototransistores tienen la ventaja de permitir la amplificación de la señal, pero están limitados en frecuencia y son sensibles a cambios de temperatura. Los rectificadores controlados de silicón, (SCR, SILICON CONTROLLED RECTIFIERS) actúan como interruptores de corriente cuando son expuestos a la luz.

Cuando la fuente de luz es una lámpara incandescente, un filamento ligero y rígido es usado para soportar mejor los esfuerzos mecánicos y la vibración. De esta forma la vida del bulbo es alargada, permitiendo un ensamblaje no reemplazable del bulbo para estar permanentemente montado con el codificador.

Las fuentes de luz de estado sólido son de muy larga vida y soportan esfuerzos mecánicos, pero ofrecen una luminiscencia menor que el bulbo incandescente.

El desempeño de los codificadores de contacto, magnéticos o capacitivos está limitado por las dimensiones prácticas de los sensores o de los segmentos del disco codificado. Por otra parte, los codificadores ópticos están limitados por la precisión del ensamblaje mecánico. Los codificadores de tipo ópticos son generalmente mucho más caros y de mejor desempeño, alcanzando una vida útil más amplia. Por este motivo, son usados con mejores resultados en condiciones de alta velocidad, resolución y precisión.

3.2.3. Clasificación por tipo de codificación

Dependiendo del tipo de codificación que realizan, existen dos clases de codificadores: absolutos e incrementales. Los codificadores absolutos entregan como salida un código que representa la posición absoluta del disco. Los codificadores de tipo incremental entregan una serie de pulsos; contando el número de pulsos es posible determinar la posición relativa del disco con respecto al punto de inicio del desplazamiento. A pesar de su versatilidad tienen la deficiencia de detectar el movimiento en una sola dirección; por lo que en movimientos para varias direcciones es necesario el contar con un arreglo de codificadores.

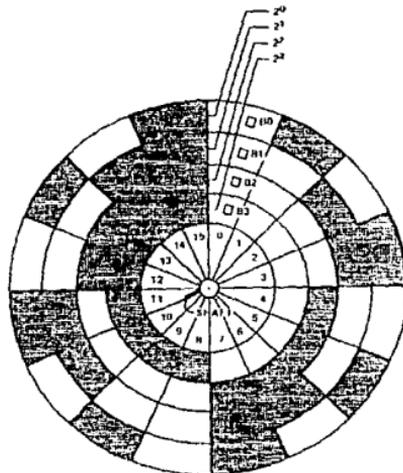
Sin importar el método usado para su lectura, los codificadores difieren en el tipo de patrón de código que utilizan. En los codificadores absolutos el patrón empleado representa la posición angular absoluta del eje y la salida es la palabra digital que lo representa, como lo muestra la fig. #14.

En el caso de los codificadores incrementales, el patrón es una secuencia uniforme de segmentos espaciados con igual incremento de ángulo. De esta forma la señal de salida se comporta como un tren de pulsos. Al contar estos pulsos,

se puede determinar la posición relativa al punto de inicio del conteo. Como este punto de inicio de conteo no tiene una posición fija en el disco se conoce como "cero flotante". La desventaja de usar un decodificador con un "cero flotante" es que si por alguna razón se pierde la cuenta, como puede ocurrir en una falla eléctrica o una falla de transmisión, el error se acumulará hasta que se reinicie la cuenta. Para evitarlo, se añade comúnmente una marca de referencia en algún punto del disco que nos permite garantizar cuando el disco ha completado una vuelta y en este punto reiniciar el conteo. De esta forma cuando suceda un error, este persistirá hasta que se encuentre la referencia de reinicio de conteo. En este punto la cuenta se reiniciará y el problema se habrá solucionado.

Los codificadores incrementales sólo requieren de una señal que indique el incremento del ángulo de giro. El conteo se puede realizar con un circuito digital sencillo y que no consuma mucha potencia. Altas resoluciones pueden ser acomodadas en elementos pequeños, sobretodo si

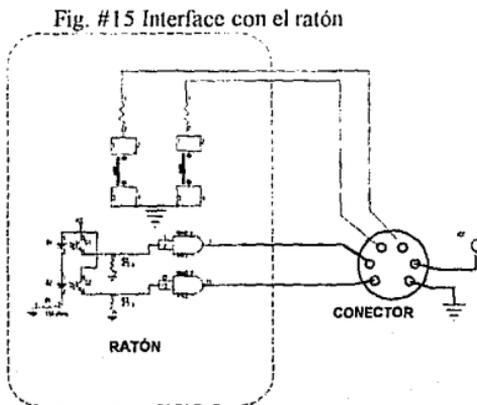
Fig. #14 Codificación del disco en un codificador de palabra absoluta



se trata de codificadores ópticos. Para lograr la determinación del sentido se utilizan dos sensores, por lo que se manejan dos cuentas al mismo tiempo, lo que implica contar con dos líneas de datos. La estructura exacta de estos codificadores se discute con más detalle en la sección E, cuando se hable del acondicionamiento de las señales.

3.3. Características del dispositivo apuntador.

El sistema de medición quedo formado por un dispositivo apuntador del tipo de puerto asignado, "bus mouse", que presenta las señales de los sensores a su salida; de esta forma el sistema es directamente conectado al dispositivo y se garantiza que siempre se pueda conocer el estado del dispositivo, además da libertad en la manipulación e interpretación de los datos. El dispositivo cuenta con dos codificadores angulares de movimiento, de tipo óptico incremental, para seguir un movimiento en dos direcciones. Cada uno de los codificadores dos sensores (como se muestra en la fig.#15), por lo que se puede determinar el sentido del movimiento, lo que nos permite determinar el valor absoluto del movimiento, como será explicado en la siguiente sección. El "ratón" cuenta además con dos interruptores para dar entrada a comandos.



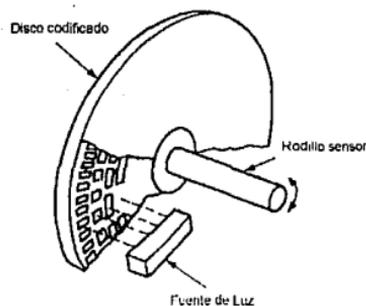
Los detectores se encuentran alineados a 90° uno respecto al otro, haciendo contacto con la bola metálica por medio de rodillos. Estos transmiten el movimiento que realiza el ratón a los codificadores. El dispositivo cuenta con un circuito CMOS 4093 (compuerta lógica NAND de entrada "Schmitt trigger") para acondicionar y cuadrar la señal.

3.4. Acondicionamiento y procesamiento de señales.

Para la correcta interpretación de las señales del dispositivo es necesario conocer más ampliamente como son generadas.

El principio de operación de un codificador incremental es la generación de una forma de onda simétrica y repetitiva capaz de ser usada para monitorear el movimiento. Los componentes del codificador incremental son: Fuente de luz, disco codificado, (que funciona como "gatillo" para disparar el proceso), sensor de luz, electrónica de acondicionamiento de señal y un ensamble a la fuente de movimiento, como se muestra la fig. #16

Fig.#16 Disparo en el codificador óptico



La entrada mecánica acciona el disco que, al cortar el haz, modula la intensidad de la luz que incide en el sensor. La salida del sensor es función

de la intensidad de luz. Esta modulación de la señal lumínica genera las siguientes formas a la salida del sensor:

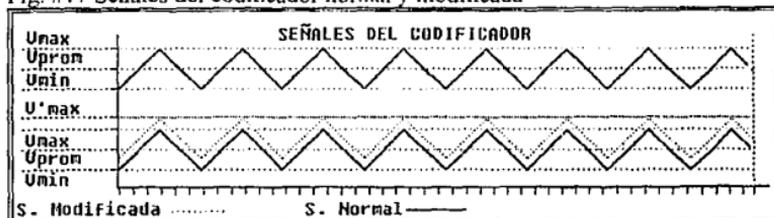
- Onda senoidal.
- Onda cuadrada.
- Tren de pulsos.

La forma básica del "gatillo" es la de una rejilla óptica en un disco; está formada por líneas y espacios alternados y del mismo espesor. Cuando el disco se mueve con respecto a la fuente de luz, se corta el haz; con lo que la luz que incide en el sensor aumenta y baja de intensidad sucesivamente. En la práctica, se coloca una retícula; esta permite el paso de toda la luz útil para sensar y evita que se filtre luz cuando son los períodos de oscuridad. Además, el uso de la retícula facilita la construcción, al evitar que el sensor se tenga que alinear en forma precisa con respecto al haz. Otra ventaja de la retícula es que las imperfecciones del disco (como cortes, agujeros y partículas de polvo) no afectan significativamente, al ser la salida un promedio de la luz que incide en varias líneas.

En la práctica, aún con el uso de la retícula no se puede alcanzar un corte total de luz; ya que existen espacios entre sensor, retícula y disco por donde esta alcanza a pasar.

Teóricamente, la forma de onda de la señal de salida se considera triangular (fig. #17); ya que idealmente el sensor tiene un respuesta lineal (aunque en la práctica la señal es casi senoidal). La amplitud máxima (V_{max}) sucede cuando la retícula esta completamente abierto y depende de

Fig. #17 Señales del codificador normal y modificada



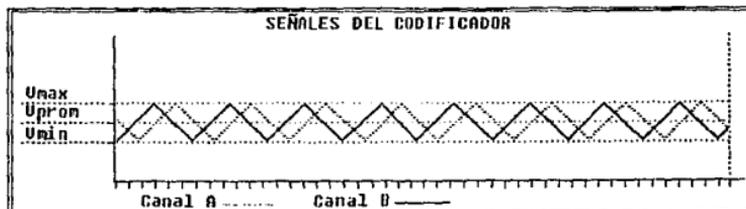
sensibilidad del sensor. La amplitud mínima, V_{min} , depende de la cantidad de luz que alcanza a filtrarse cuando el disco cortador del haz se encuentra cerrado. En la práctica, el voltaje pico a pico es la componente útil de la señal de salida y está limitado por la efectividad del mecanismo de disparo.

La forma más común de producir la salida codificada es generando un pulso cada vez que la señal de salida pasa a través del voltaje promedio (V_{prom}). Idealmente, los cruces de la señal de salida con la línea imaginaria del voltaje promedio están espaciados a la misma distancia y corresponden al movimiento de un segmento del disco. El resultado es una serie de pulsos espaciados a la misma distancia; con los que se puede monitorear, en forma precisa, la entrada mecánica al codificador.

El principal problema de este arreglo es que la señal, puede sufrir un corrimiento a causa de cambios en el nivel de voltaje de excitación de la fuente de luz, o por variaciones en la sensibilidad del sensor de luz (señal modificada en la fig. #17). Cuando esto ocurre, los pulsos generados (cuando se cruza por el nivel promedio de la señal) ya no son válidos, pues no están separados a la misma distancia.

En la práctica, los codificadores están formados por dos canales cuyas señales están defasadas 90° . Esto nos da la capacidad de alcanzar una resolución de 4 veces el número de segmentos que conforman el disco. Esta capacidad nos duplica la resolución del sistema a un bajo costo. Como muestra la figura #18, las 2 señales (que representan a los dos canales) se encuentran defasadas 90° ; al detectar los cruces, somos capaces de detectar un movimiento equivalente a la mitad del ancho de una de las líneas.

Fig.#18 Defasamiento de las señales en los canales

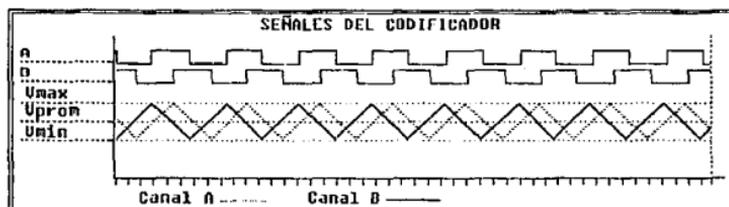


Por ejemplo, si nuestro disco cuenta con 24 líneas y 24 espacios, entonces seremos capaces de detectar movimientos de:

$$\left(\frac{360 \left(\frac{\text{grados}}{\text{revolucion}} \right)}{4 \left(\frac{\text{cuentas}}{\text{segmento}} \right) \times 24 \left(\frac{\text{segmentos}}{\text{revolucion}} \right)} \right) = 3.75 \left(\frac{\text{grados}}{\text{cuenta}} \right)$$

La utilización de este arreglo permite, además, la determinación del sentido del movimiento. Se puede establecer la convención (como se indica en la figura#19) de que cuando el disco gira en sentido positivo, el haz de luz que es cortado primero es el del canal "A" y luego, al recorrer medio sector, es cortado el del canal "B". Por otro lado, cuando el disco gira en sentido contrario el que se corta primero es el del canal "B" y luego el "A".

Fig.#19 Generación de los pulsos a partir de los canales del ratón

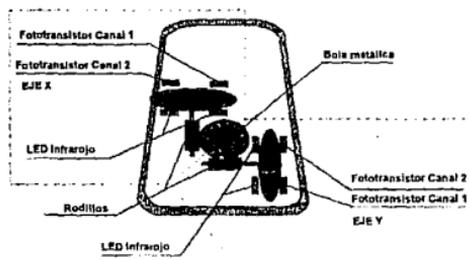


Si tomamos el canal "A" como referencia, el signo del defasamiento de las señales generadas nos indica el sentido del movimiento. De esta forma, si el canal "B" se encuentra defasado a -90° entonces el movimiento se realiza hacia la izquierda. Cuando el defasamiento es de $+90^\circ$ el movimiento se realiza a la derecha.

3.5. Características del decodificador en el dispositivo apuntador.

El codificador óptico del ratón, que se empleó en este diseño, esta compuesto por un rodillo, el disco codificado con 24 secciones y dos pares de arreglos LED-RETICULA-FOTOTRANSISTOR como se muestra en la fig. #20.

Fig.#20 Descripción del ratón



Si tomamos en cuenta que el rodillo que detecta el movimiento es de un diámetro de 0.1484 pulg., entonces la resolución de nuestras mediciones esta dada por la fórmula:

$$\text{Resolución} = \left(\frac{\text{Perímetro del rodillo}}{\left(\frac{\text{cuentas}}{\text{segmento}} \right) \times \left(\frac{\text{segmentos}}{\text{perímetro}} \right)} \right) =$$

$$\left(\frac{\pi \times 0.15 \left(\frac{\text{pulgadas}}{\text{revolución}} \right)}{4 \left(\frac{\text{cuentas}}{\text{segmento}} \right) \times 24 \left(\frac{\text{segmentos}}{\text{revolución}} \right)} \right) = 0.0049 \left(\frac{\text{pulgadas}}{\text{cuentas}} \right) = \frac{1}{200} \left(\frac{\text{pulgadas}}{\text{cuentas}} \right)$$

IV. ARQUITECTURA DEL SISTEMA.

INTRODUCCIÓN.

En este capítulo se describe la forma en que fue organizado el sistema de procesamiento y la interface con el dispositivo apuntador, ya que de esto depende su funcionalidad y eficiencia.

Primeramente en la sección 1) **Elección de la arquitectura del sistema**, se detallan las características con las que cuenta el sistema y que determinaron el uso de una cierta tecnología.

En la siguiente parte, 2) **Descripción de la arquitectura del equipo**, se analiza el sistema en 2 partes: 1) Descripción del sistema mínimo de procesamiento, en el que microprocesador es el centro de la arquitectura del sistema, de su mapa de memoria depende la forma de organización y la forma en que fueron enlazados los distintos dispositivos externos. y 2) Descripción de la tarjeta de aplicación, que incluye el teclado de comandos, la fuente de alimentación y la interface con el dispositivo apuntador.

Se han incluido en un apéndice las características de los dispositivos MCS'51, haciendo énfasis en la forma en que esta organizada su memoria, así como, se describe la organización del circuito impreso de la tarjeta de procesamiento y de aplicación.

4.1. Elección de la arquitectura del sistema.

Como ya se ha mencionado, se buscó como objetivo el diseñar un sistema de procesamiento digital de bajo costo y que fuera fácilmente adaptable a distintas aplicaciones. Se eligió el uso de un microcontrolador por la facilidad de manipulación y almacenamiento de datos que nos ofrece; además de contar internamente con una serie de dispositivos, como son: puertos serial y paralelo, memoria tipo RAM, contadores de 16 bits y manejo de fuente de poder (Power management).

Fue elegida la familia de microcontroladores MCS'51, de la compañía INTEL, por ofrecernos todas las ventajas anteriores a un costo reducido y por su disponibilidad.

Las principales características de la familia MCS'51 son:

1. Microprocesador de 8 bits optimizado para aplicaciones de control.
2. Líneas bidireccionales de entrada/salida, individualmente direccionables .
3. Un transmisor-receptor universal asíncrono [UART] tipo "full-duplex".
4. Vectores de interrupción de 6 posibles fuentes con dos niveles de prioridad.
5. Generador de reloj interno.
6. Memoria para programas de hasta 64 Kbytes.
7. Memoria interna de programas.
8. Memoria interna tipo RAM.
9. Contadores/relojes de 16 bits.
10. Set de instrucciones para procesamiento Booleano (Operaciones lógicas de 1 bit).

La familia de microcontroladores MCS'51 cuenta con varios dispositivos, de distintas capacidades, para satisfacer diversos requerimientos. Las características de los miembros de la familia MCS'51 se describen en la siguiente tabla:

Características de los microcontroladores de la familia MCS'51

| Nombre del dispositivo | Versión sin ROM | Versión con EPROM | Bytes en ROM | Bytes en RAM | Contadores de 16 bits | Tipo de Tecnología |
|------------------------|-----------------|-------------------|--------------|--------------|-----------------------|--------------------|
| 8051 | 8031 | 8751 | 4K | 128 | 2 | HMOS |
| 8051AH | 8031AH | 8751H | 4K | 128 | 2 | HMOS |
| 8052 | 8032AH | 8752BH | 8K | 256 | 3 | HMOS |
| 80C51AH | 80C31BH | 87C51 | 4K | 128 | 2 | CHMOS |

De ellos decidimos utilizar el 80C31BH por ser de un costo menor; ya que no tiene ROM interna (que requiere ser grabada de

fábrica para ser funcional). En cambio tiene capacidades de ahorro de energía, por lo que son aptos para sistemas portátiles, como es el caso del Ictiómetro.

El sistema (ver fig. #21) está diseñado para manejar una memoria de programa por hasta 8 Kbytes, pero para la aplicación del Ictiómetro se usó una memoria de 4 Kbytes. La memoria de programa está implementada en EPROM, de tecnología CMOS, para garantizar un mínimo consumo de energía.

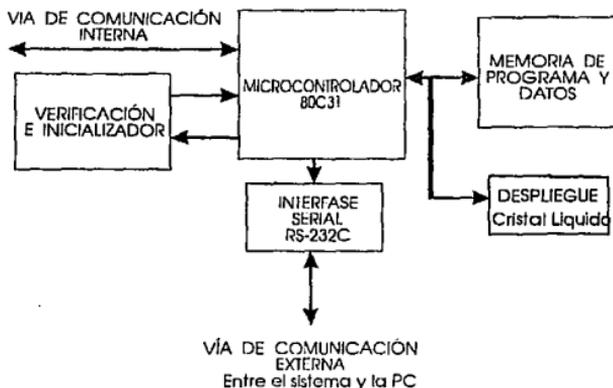
La memoria de datos tiene una capacidad de 64 Kbits (8 Kbytes), con lo que se pueden almacenar hasta 4 mil mediciones (si cada una ocupa 2bytes). Esta se implantó con un dispositivo EEPROM (ELECTRICAL ERASABLE PROGRAMMABLE READ ONLY MEMORY) XL2865 de la compañía EXCEL; la cual tiene la capacidad de ser borrada y programada con un voltaje de 5V. De esta forma los datos son almacenados de manera no-volátil, por lo que la información se mantiene aunque el circuito se apague.

Otras características de la memoria de datos [21] son:

- Tiempo de acceso de 250 ns.
- Operación a 5 V, incluyendo escritura.
- Ciclo rápido de escritura no-volátil:
 - Operación automática de escritura fuera de ciclo.
 - Captura interna de datos.
 - Captura interna de direcciones.
 - Borrado automático de datos antes de la escritura.
 - Poleo de datos como indicador de estado.
 - Terminal de estado LISTO/OCUPADO.
- Escritura automática por página.
- Protección contra falsa escritura.
- Entradas compatibles con TTL.
- Hasta 10,000 re escrituras por byte.
- Retención del dato por hasta 10 años.

Para la interface fue necesario usar, RS-232, un circuito que no necesitara una alimentación de ± 12 V. El ICL232 de INTERSIL [11] está diseñado para operar con niveles bajos de voltaje; integrando en el mismo dispositivo el manejador de línea, tanto de transmisión como de recepción.

Fig.#21 Diagrama de bloques del sistema mínimo



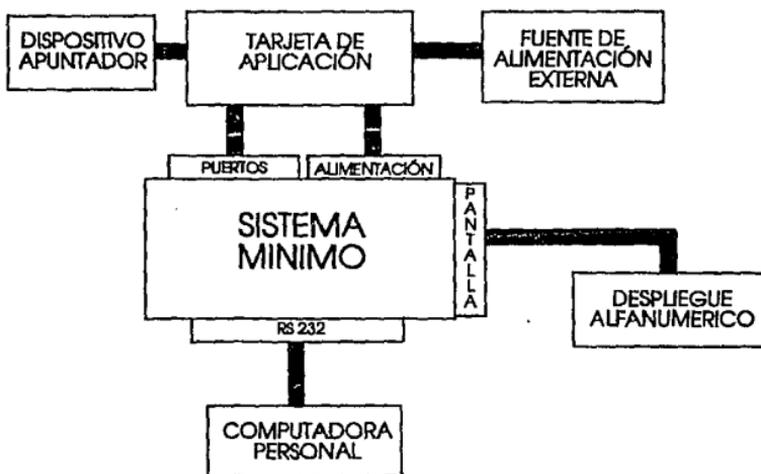
Como parte de los requerimientos del equipo de medición era necesario algún tipo de despliegue, que mostrara el resultado de las mediciones. Por esto, se decidió que el sistema mínimo tuviera integrada la capacidad de mostrar datos; con la ventaja de que esta queda disponible para cualquier otra aplicación del sistema. El uso de pantallas con base en LED's no es recomendable para equipos portátiles, debido a su gran consumo de energía. Las pantallas de cristal liquido (LCD, LIQUID CRYSTAL DISPLAY) consumen menos energía, pero son mucho más difíciles de manejar ya que requieren de circuitos osciladores y de un constante refresco de información. Para reducir el problema, la opción más viable es el uso de una pantalla con un sistema de control integrado. Se escogió una pantalla de cristal liquido con controlador integrado tipo AND 491 [12]. Esta pantalla tiene 2 renglones, es capaz de desplegar hasta 32 caracteres

alfanuméricos al mismo tiempo y tiene la ventaja de operar en forma totalmente autónoma, controlando el proceso de despliegue; por lo que sólo es necesario enviarle instrucciones para que ella opere.

Para reducir el espacio necesario para la construcción del sistema y lograr un ahorro de energía, se optó por el uso de Arreglos Lógicos Programables (PAL, PROGRAMMABLE ARRAY LOGIC) para implementar la lógica de soporte del sistema. Se escogió, de la familia de PAL's de National Semiconductor, el circuito PAL20L8 [24], este dispositivo se caracteriza por permitir implementar hasta 8 funciones con 20 posibles señales de entrada.

Para realizar la aplicación del sistema mínimo como equipo de medición de pescados (fig.#22), se implementó esta parte lógica en otro PAL20L8, con el fin de ocupar el menor espacio posible. También se usaron dispositivos discretos, como filtros de la señal.

Fig.#22 Diagrama de bloques de lctiómetro



4.2. Descripción de la Arquitectura del equipo

El sistema fue concebido como una arquitectura abierta, que diera la oportunidad de usar el sistema en otras aplicaciones. Por tanto, se pensó en separarlo físicamente en dos partes: la tarjeta de procesamiento y la tarjeta de aplicación. La primera de ellas contendría el sistema mínimo, compuesto por el microcontrolador y la lógica de soporte para su operación. La segunda agruparía a los sistemas necesarios para la operación del equipo de medición de la longitud de peces (Ictiómetro): la lógica de acondicionamiento e interface con el ratón y la fuente de poder.

En esta sección se describe cada una de estas tarjetas. De la primer tarjeta se analiza su memoria, comunicaciones, despliegue y seguridad, y de la tarjeta de aplicación, la interface con el sensor, el teclado de comandos y la fuente de poder

4.2.1. Descripción del sistema mínimo de procesamiento.

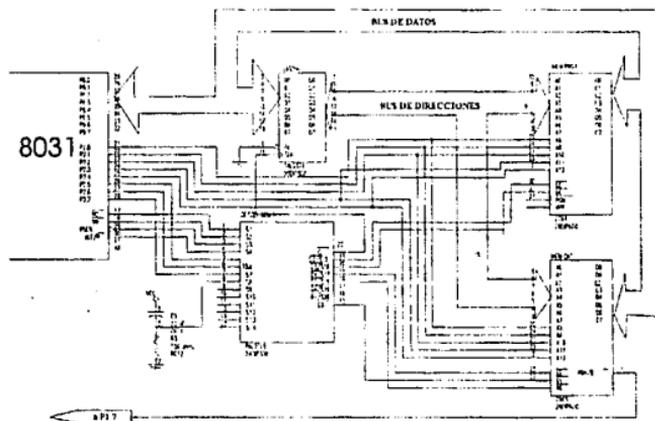
El Sistema mínimo de procesamiento, mostrado en la figura #21, cuenta con las siguientes características:

1. Memoria de programas en EPROM de 8 Kbytes.
2. Memoria de datos en EEPROM de 8 Kbytes.
3. Lógica de decodificación en PAL.
4. Circuito de protección ("Watch dog").
5. Conexión con la pantalla de cristal liquido AND491
6. Interface para comunicaciones con la especificación EIA RS-232 por 3 hilos.

4.2.1.1. Memoria.

La familia MCS'51 tiene la característica de que su bus de datos y la parte baja de las direcciones se encuentran multiplexadas, es decir, que comparten las mismas líneas de entrada al circuito. Es necesario separar ambos tipos de señales para poder manejarlas. Esto se resolvió de la forma como se muestra en la figura #23. Para lograr la multiplexión del bus de datos y la parte baja de la dirección de memoria se usó la señal de ALE (Address Latch Enable). Con un circuito retención (LATCH) 74LS124 de 8 bits, se mantiene la parte baja de la dirección de memoria, utilizando a ALE como señal de disparo para efectuar la captura del dato.

Fig.#23 Esquema general de la arquitectura



Por otra parte, los dispositivos de la familia MCS'51 tienen una arquitectura tipo Harvard; ya que mantienen dividido el mapa de almacenamiento de datos (MAD) y de programa (MAP), reservándole a cada uno señales de control para realizar la lectura y escritura de datos. Para programas se implementó la parte baja del MAP con una memoria EPROM 2732 [23], ocupando las direcciones 00000H a la

01FFFH. El diseño se realizó para memorias de 64 Kbits (8kbytes x 8); pero se hizo compatible con memorias de 32 Kbits (4kbytes x 8), de modo que para otras aplicaciones se puede usar cualquiera de las dos. Cuando son usadas las memorias de 32 Kbits se ocupa la zona comprendida entre las direcciones 00000H a 00FFFH, con repeticiones de la información entre las líneas y entre 01000H a 01FFFH; lo que implica que la memoria puede ser accesada en cualquiera de las direcciones

.En forma esquemática el mapa de memoria del sistema es mostrada en la tabla siguiente:

MAPA DE MEMORIA DEL SISTEMA

| DIRECCIÓN | MAPA DE ALMACENAMIENTO DE PROGRAMA. | MAPA DE ALMACENAMIENTO DE DATOS. |
|-----------|-------------------------------------|----------------------------------|
| 0000H | CÓDIGO DEL PROGRAMA | ALMACENAMIENTO DE DATOS |
| 0A00H | TABLA DE DATOS DE LA PANTALLA | |
| 1000H | EXPANSIÓN A 64 Kbytes | |
| 2000H | No usada | PANTALLA DE DESPLIEGUE |
| 3000H | | No usada |
| 4000H | | CÓDIGO DE PROGRAMA |
| 4A00H | | TABLA DE DATOS DE LA PANTALLA |

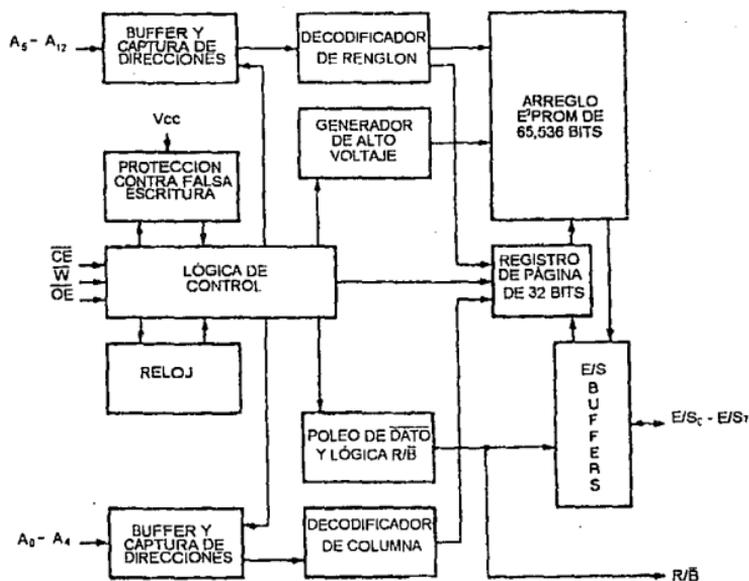
Nota: el lugar que ocupa la pantalla dentro del mapa de memoria será explicada en la sección de despliegue .

Para el sistema era indispensable contar con una zona de la memoria donde se pudieran almacenar de forma permanente los datos que utiliza; específicamente la información que se despliega en la pantalla y que sirve de interface con el usuario. Aunque el microcontrolador permite el acceder partes de la memoria de programa como tablas de datos, su direccionamiento es en forma indexada. Por esta causa fue preciso direccionar a la memoria de

programa dentro del MAD. En la zona comprendida entre las direcciones 04000H a la 05FFFH del MAD, se encuentra direccionada la memoria de programa. De esta forma, el acceso es más versátil ya que al usar las instrucciones dedicadas al acceso de datos se cuenta con la capacidad de realizar escrituras y lecturas.

La memoria de datos fue implementada con un circuito EEPROM XL2865 [21] y ocupando la parte baja del MAD, en la figura #24 se muestra su estructura interna. Este tipo de memorias cuenta con la señal Listo/ocupado (R/\bar{B}) que indica cuando el ciclo de escritura interna ha concluido, para poder garantizar que los datos han sido almacenados. Dicha señal está directamente conectada a la terminal 7 del puerto 1; de esta forma su estado puede ser permanentemente revisado.

Fig.#24 Estructura interna del circuito de memoria EEPROM XL2865



Este tipo de memorias es susceptible al ruido en las señales de control. Cuando se conecta la alimentación de energía, se suceden estados de transición que la memoria puede interpretar como escrituras; aún a pesar de que la memoria cuenta internamente con protección contra falsa escritura, provocando que la información se dañe. Para evitar este problema, se decidió añadir un circuito que mantuviera deshabilitada la señal de escritura de la memoria durante algunas fracciones de segundo; entonces, cuando ya se ha asegurado que las condiciones son estables, se puede proceder a habilitar la señal. El circuito que mantiene el estado por un período de tiempo es el que carga al capacitor. Este período está dado por el inverso del producto de la resistencia y capacitancia, mucho mayor que el tiempo que tarda en ejecutarse la inicialización interna del microcontrolador. Con esto se evita que el ruido del encendido se interprete como escritura.

Todas las funciones lógicas fueron implementadas en un circuito PAL 20L8 de National Semiconductor el cual es un arreglo lógico programable con la capacidad de 8 salidas negadas y 20 entradas. Cada salida es la negación de la suma de 8 términos mínimos, los cuales pueden ser función de cualquiera de las 20 señales de entrada o de las salidas. Las funciones implementadas son:

$$\text{CLK} = \overline{\text{ALE}}$$

$$\overline{\text{CE}}_{\text{prog}} = \overline{\text{A13}} \cdot \overline{\text{A14}} \cdot \overline{\text{A15}} + \overline{\text{A13}} \cdot \overline{\text{A14}} \cdot \overline{\text{A15}}$$

$$\overline{\text{OE}} = \overline{\text{PSEN}} \cdot \overline{\text{A13}} \cdot \overline{\text{A14}} \cdot \overline{\text{A15}} + \overline{\text{RD}} \cdot \overline{\text{A13}} \cdot \overline{\text{A14}} \cdot \overline{\text{A15}}$$

$$\overline{\text{CE}}_{\text{dat}} = \overline{\text{A13}} \cdot \overline{\text{A14}} \cdot \overline{\text{A15}}$$

$$\overline{\text{WE}} = \overline{\text{ENC}} \cdot \overline{\text{P1.5}} \cdot \overline{\text{WR}}$$

$$\overline{\text{R/W}}_{\text{dis}} = \overline{\text{A13}} \cdot \overline{\text{A14}} \cdot \overline{\text{A15}}$$

$$\text{E}_{\text{dis}} = \overline{\text{WR}}$$

$$\overline{\text{OE}}_{\text{dis}} = \overline{\text{RD}} \cdot \overline{\text{A13}} \cdot \overline{\text{A14}} \cdot \overline{\text{A15}}$$

Donde:

An es el bit *n* del bus de direcciones.

4.2.1.2. Comunicaciones.

Una de las características con que el sistema mínimo debe contar es el compartir información con otros sistemas de una forma sencilla y estándar. Dos de los métodos más usuales de comunicación entre dispositivos digitales son la comunicación serial y la comunicación en paralelo. Para ambos tipos se han desarrollado estándares y protocolos, con el fin de que sea su acceso sea universal. La comunicación en paralelo permite una transmisión de información más rápida, al ser realizada por medio de varias vías simultáneamente. Es la forma común de acceder distintos tipos de sistemas periféricos. Sin embargo, el puerto paralelo de las computadoras tipo PC-Compatibles no cuenta con un bus de datos bidireccional, debido a que su principal uso es para comunicarse con la impresora; es decir, las líneas de datos sólo funcionan en un solo sentido que es el de enviar datos. Para comunicar información en sentido inverso, sería necesario el uso de las señales de control del intercambio de información (Handshake); con el inconveniente de que únicamente son 5, por lo que no es posible transmitir un Byte al mismo tiempo. Esto limita el uso genérico del puerto paralelo de las PC's.

La comunicación serial nos ofrece ventajas de implantación, ya que en nuestra aplicación y luego de reducir el estándar solamente son necesarias tres líneas de comunicación para establecer el enlace: transmisión, recepción y tierra. Originalmente, el puerto de comunicación serial en una computadora PC sigue la norma EIA (Electronic Industries Association) RS-232 [4], el cual define 25 líneas para el manejo de los datos, el control del tránsito de las señales y hasta un segundo canal de comunicaciones. Esta norma se definió pensando en comunicaciones a media distancia conectando directamente las dos máquinas; o a larga distancia, con el uso de un Equipo de Comunicación de Datos (DCE, Data Communication Equipment). Sin embargo, no todas las líneas son indispensables para realizar una comunicación; por lo que en la mayoría de las

computadoras PC únicamente han sido implementadas 7 líneas que son las siguientes:

Señales de la interface RS232

| NOMBRE DE LA SEÑAL | NO. DE TERMINAL | FUNCIÓN |
|--------------------|-----------------|-----------------------------|
| TXD | 2 | Transmisión de datos |
| RXD | 3 | Recepción de datos |
| RTS | 4 | Requerimiento de envío |
| CTS | 5 | Listo para envío |
| DTR | 20 | Terminal de datos lista |
| CD | 8 | Detector de señal portadora |
| GND | 7 | Tierra de la señal |

Estas señales de intercambio de información son utilizadas para el control de un MODEM (modulador/demodulador), el estado del canal de comunicación y el intercambio de datos (Handshake). Debido a que la comunicación del sistema mínimo con la PC se realiza a corta distancia se implementó un enlace entre equipos Terminales de Datos (DTE, Data Terminal Equipment). De esta forma ambas máquinas se encuentran dedicadas a la comunicación, sin existir momentos en los que esta no se pueda realizar; Disminuyendo las líneas necesarias para la comunicación.

Para reducir de manera efectiva el número de líneas usadas para el enlace y no tener que asignar señales de los puertos del microcontrolador para su operación, se hizo una evaluación de las funciones de la interface. El tipo de comunicación que se decidió usar es asíncrono; por lo que la detección del dato se lleva a cabo por medio de bits de inicio y paro (función que es realizada internamente por el puerto). A pesar de esto, la computadora espera las señales CTS y CD para iniciar la búsqueda del dato y continuar con la transmisión. Con el objeto de que el puerto siempre se encuentre listo para transmitir o recibir datos, estas señales fueron directamente alambradas en el conector a las señales RTS y DTR

respectivamente, que se activan automáticamente por el BIOS [6] de la computadora.

Las computadoras que se revisaron añadían una línea más al manejo de las señales, llamada "Dato listo" *DSR* (data set ready). Esta última línea fue conectada con *DTR* y *CTS*. De esta forma se redujo la conexión de los equipos a 3 líneas: *GND*, *TXD*, *RXD*.

El sistema mínimo se consideró como una terminal de datos, por lo que decidimos usar la configuración de terminales que define el norma RS-232 para un conector tipo DB-9P en las computadoras PC. De acuerdo al estándar, se tiene la siguiente configuración de terminales:

| TERM. | E/S | SEÑAL EN PC | SEÑAL EN ICTIÓMETRO |
|-------|-----|-------------------------|----------------------|
| 1 | E | Detector de portadora | No conectada |
| 2 | E | Recepción de Datos | Recepción de Datos |
| 3 | S | Transmisión de Datos | Transmisión de Datos |
| 4 | S | Terminal de datos lista | No conectada |
| 5 | NA | Tierra | Tierra |
| 6 | E | Datos listos | No conectada |
| 7 | S | Solicitud de envío | No conectada |
| 8 | E | Listo para envío | No conectada |
| 9 | E | Indicador de timbre | No conectada |

Donde:

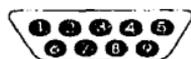
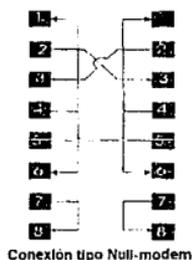
E= Entrada

S= Salida

NA= No aplicable

Se decidió que las conexiones entre las terminales que se unen fueran hechas directamente en el cable de conexión, permitiendo su uso en otras aplicaciones donde se realice comunicación entre equipos DTE. De esta forma la conexión necesaria en el cable tiene la configuración mostrada en la figura #25.

Fig.#25 Diagrama de conexiones para comunicación "null modem"

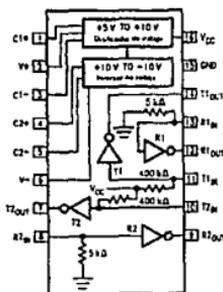


Disposición de terminales en el DB-09

El standard RS-232 define como valor lógico "alto" o "1" a un voltaje bajo carga entre -3 V y -15 V. Mientras que un valor lógico "bajo" o "0" es representado con un voltaje entre +3 V y +15 V. Para poder ser compatibles, es necesario transformar los niveles lógicos que maneja el sistema (0 V y 5 V) a los que requiere RS-232. Para lograrlo comúnmente son usados los dispositivos MC1488 (en la señal de transmisión) y MC1489 (en la recepción) para hacer la transformación de voltajes. Sin embargo, el MC1488 requiere de una alimentación de ± 12 V mientras que el sistema se limitó al uso de una sola fuente de energía (a 5 V); por lo que no es posible el uso de estos dispositivos. Se buscaron otras posibilidades y, se decidió utilizar el ICL232 de Intersil [11] (mostrado en la fig. #26), que es un circuito integrado diseñado específicamente para cumplir los requerimientos del EIA RS-232. Este requiere de una alimentación

única de +5 V para operar; ya que cuenta internamente con dos convertidores de voltaje a +10 V y a -10 V.

Fig.#26 Circuito ICL232



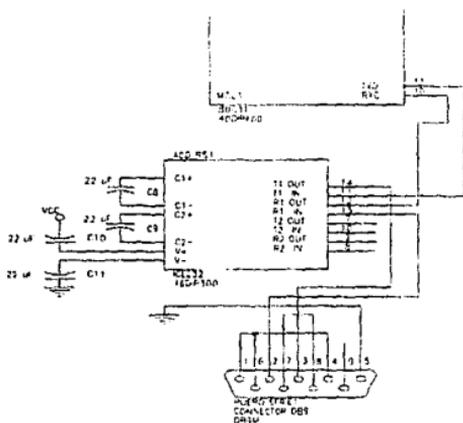
Las características de este dispositivo son:

- Compatible con todas las especificaciones de RS-232C.
- Requiere una sola fuente de +5V.
- Contiene un convertidor de voltaje interno.
- Bajo consumo de energía.
- 2 transmisores con las siguientes características:
 - ±9 V de salida para entradas con niveles TTL.
 - 300 Ω de impedancia sin fuente de poder.
 - Limitador de salida de corriente.
 - Entradas compatibles con TTL o CMOS.
- Y 2 Receptores:
 - ± 30 V de rango de voltaje de entrada.
 - De 3 a 7 kΩ de impedancia de entrada.
 - 0.5 V de histéresis para reducir errores por ruido.

Para su utilización, el ICL232 requiere de cuatro capacitores que le permiten realizar la función de multiplicación de voltaje. La literatura se recomienda el uso de capacitores de 22 μF. Para realizar la conversión, el circuito utiliza dos fuentes de carga, durante dos fases

de un reloj interno. La interface fue implementada de la forma que indica la figura #27.

Fig.#27 Conexión del ICL232



4.2.1.3. Despliegue.

Se eligió la pantalla AND 491 (su estructura interna se muestra en la fig. #28) de cristal líquido por sus capacidades, entre las que podemos mencionar [12]:

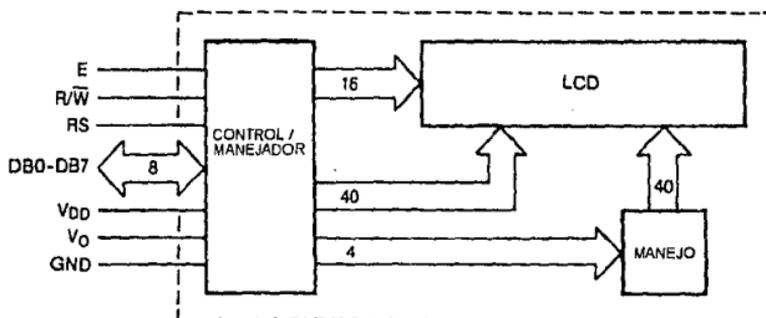
- Sistema de control interconstruido.
- Memoria RAM para el despliegue.
- Generador de caracteres en ROM.
- Formato de caracteres de 5 x 7 puntos.
- Interface directa con procesadores de 4 u 8 bits.
- 11 comandos de control.
- Operación con una fuente de poder de +5 V.

- Rango amplio de temperaturas de operación (0°C a $+50^{\circ}\text{C}$).

La pantalla puede desplegar 2 líneas de 16 caracteres cada una. Además, almacena en memoria interna 160 tipos de caracteres, números, símbolos o ideogramas; con la posibilidad de implementar 8 caracteres más, de acuerdo a las necesidades del usuario.

Fig.#28 Organización interna de la pantalla AND491

DIAGRAMA DE BLOQUES



La pantalla AND 491 cuenta con 14 terminales para realizar la conexión. La asignación de terminales corresponde a la siguiente tabla:

| TERM. NO. | SEÑAL | FUNCIÓN |
|-----------|-----------------|---|
| 1 | GND | 0 V |
| 2 | V _{DD} | 5 V |
| 3 | V _O | SEÑAL DE INTENSIDAD DEL DESPLIEGUE |
| 4 | RS | "H" ENTRADA DE DATOS "L" ENTRADA DE COMANDOS |

| | | | |
|----|-----|---|---------------|
| 4 | RS | "H" ENTRADA DE DATOS "L" ENTRADA DE COMANDOS | |
| 5 | R/W | "H" LECTURA DE DATOS "L" ESCRITURA DE DATOS | |
| 6 | E | SEÑAL DE HABILITACIÓN | |
| 7 | DB0 | BUS DE DATOS | |
| 8 | DB1 | | |
| 9 | DB2 | | |
| 10 | DB3 | ENLACE 4 BITS | ENLACE 8 BITS |
| 11 | DB4 | | |
| 12 | DB5 | | |
| 13 | DB6 | | |
| 14 | DB7 | | |

H = Alto

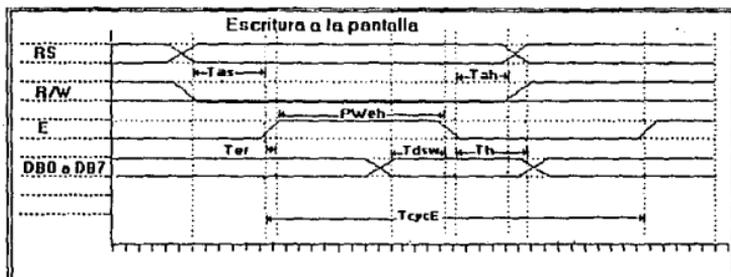
L = Bajo

Las dos primeras señales sirven de alimentación para la pantalla y fueron conectadas a la fuente de alimentación del sistema. La señal no. 3, de intensidad, sirve para controlar el voltaje que gobierna la pantalla y alcanzar un buen nivel de contraste. Este voltaje de referencia fue implementado por medio de un potenciómetro de 10 k Ω configurado como divisor de voltaje, entre la fuente de alimentación y tierra. De esta forma la intensidad del despliegue puede ser ajustado.

El resto de las líneas sirven como bus de datos y control del tráfico. La pantalla se implementó como parte del mapa de memoria de datos (MAD), reservando para su uso el área siguiente a la memoria EEPROM: las direcciones 2000H. Como la decodificación de la memoria se hizo con las 3 líneas más significativas de la dirección (A13, A14, A15), la pantalla presenta repeticiones consecutivas entre las direcciones 2000H y 21FFH. Para el manejo de la señal de selección de registros (no. 4) se utilizó la línea de dirección A0. Por lo tanto, para acceder el registro de datos es necesario hacerlo en las direcciones impares; ya sea para leer o escribir. El manejo de comandos se hace a través de las direcciones pares, al acceder el registro de instrucciones.

Las señales de escritura a memoria externa del 80C31 y las de acceso a la pantalla (mostradas en los diagramas de tiempo de la fig. #29), presentan diferencias en el orden en que son activadas.

Fig.#29 Temporización en la escritura de la pantalla

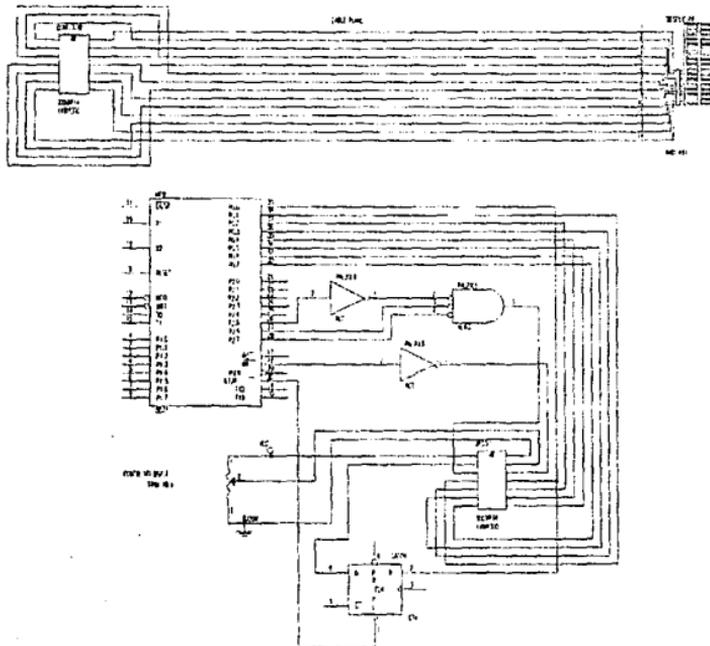


La señal de lectura/escritura de la pantalla (R/\bar{W}) debe de estar activada por un tiempo (t_s) antes de poder activar el dispositivo por medio de la señal habilitación (E). Este procedimiento es distinto al que usa el 80C31 para acceder los dispositivos periféricos; en el microcontrolador, las líneas de direcciones, las cuales son usadas para generar la señal de habilitación del circuito, se activan antes que la señal de lectura.

Para solucionar las diferencias se limitó el acceso a la pantalla solamente a escrituras; la implantación completa de la pantalla, para lograr lecturas, no se realizó ya que era una función que no se necesitaba para el sistema actual.

De esta forma la señal de habilitación es generada con la señal de escritura del microcontrolador, mientras tanto, la señal de Lectura/escritura de la pantalla (R/\bar{W}) es generada por la decodificación de las direcciones. En la fig. #30 se describen las operaciones lógicas que fueron implementadas dentro del PAL. y la conexión entre la tarjeta y la pantalla por medio de cable plano.

Fig.#30 Conexión con la pantalla. Arriba en el cable de conexión. Abajo entre el equipo y su conector



4.2.1.4. Perro guardián.

Para aumentar el nivel de seguridad del sistema se implementó un circuito de verificación de su funcionamiento. El circuito, llamado "perro guardián" (Watch Dog), verifica que el microprocesador se encuentre operando, de lo contrario genera una señal de inicialización, el objetivo de esto es que no se lleguen a afectar los datos ya almacenados en caso de que el programa pierda su secuencia.

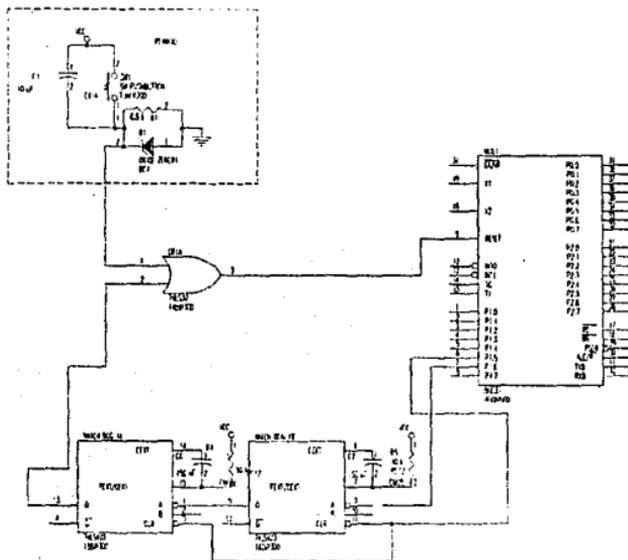
El funcionamiento del perro guardián es muy sencillo: el dispositivo es sensible a las transiciones negativas de una de las líneas de Puerto 1 (P1.6). Cuando registra una transición, es que el programa se encuentra ejecutándose en forma satisfactoria e inicia un tiempo de espera; cada vez que se sucede una transición es reiniciado este tiempo. Si por alguna razón el tiempo termina y no ha habido ninguna transición, entonces el circuito genera una señal de reinicio (reset en el microcontrolador), con lo que se restablece la operación del programa.

El 80C31 tiene la capacidad de detener su operación por largos períodos, reduciendo de esta forma el consumo de energía que requiere. En el caso de tratar de hacer esto con el "perro guardián" activado, ocasionaría la generación de una señal de reinicio al no detectarse operación del programa. Es entonces necesario poder desactivar la operación del "perro guardián" antes de detener la operación del microcontrolador.

Esta fue la razón principal para escoger al 74LS423 como monoestable, en vez de usar el común LS123; ya que es el único que nos ofrecía una señal de borrado (CLR) no conectada como señal de disparo. En los otros tipos de monoestable la señal CLR es también una señal de disparo, por lo que cualquier transición en el estado de esta señal ocasiona la generación de la señal de reinicio (reset). Cuando en el 74LS423 se activa la señal de borrado, el monoestable cambia al estado bajo sin importar la condición en la que se encuentre. Este proceso tiene que ser hecho al mismo tiempo en ambos monoestables para que no se genere ninguna señal de reinicio. Por lo que, para garantizar su activación antes de realizar el paro de operación del microcontrolador, ambas señales de borrado de los monoestables están manejadas por la terminal P1.5 del microcontrolador. Cuando ha pasado el estado de paro del controlador, se reactiva la operación del perro guardián con un nuevo disparo, asegurando así la buena operación del sistema.

Se conectaron en cascada dos dispositivos, como muestra la fig. #31. Podemos observar que cuando el primer monoestable es disparado con una transición negativa el estado de su salida cambia

Fig.#31 Perro guardián.



de bajo a alto; esto no dispara al segundo monoestable ya que éste también se acciona con transiciones negativas. Cada vez que el primer monoestable detecta una transición negativa, reinicia su estado en alto sin hacer ninguna transición. Al haber transcurrido el tiempo de duración (0.57 seg.), si no se ha vuelto a disparar, cambia su estado de alto a bajo ocasionando el disparo del segundo monoestable. Este último es el que activa la señal de reinicio.

4.2.2. Descripción de la tarjeta de aplicación

Los dispositivos adicionales al sistema mínimo para funcionar como instrumento de medición se integraron independientemente en una tarjeta de soporte. Los elementos construidos en la tarjeta de aplicación son:

- La interface con el dispositivo apuntador; así como el acondicionamiento de sus señales.
- La interface con un teclado para la entrada de comandos al sistema.
- El circuito que maneja a la pila y a una entrada externa como fuente de poder.

4.2.2.1. Teclado de comandos.

Es necesario contar con una forma de entrada de comandos al sistema, que permita al usuario ejecutar las distintas rutinas con las que trabaja el Ictiómetro. Consideramos más conveniente que la unidad de procesamiento no estuviera limitada a tener que mantener al ratón conectado para operarla, aunque el dispositivo apuntador ya cuenta con 2 botones desde los cuales se podría haber controlado toda la operación. Por esta razón se le anexó al sistema un teclado de 3 funciones.

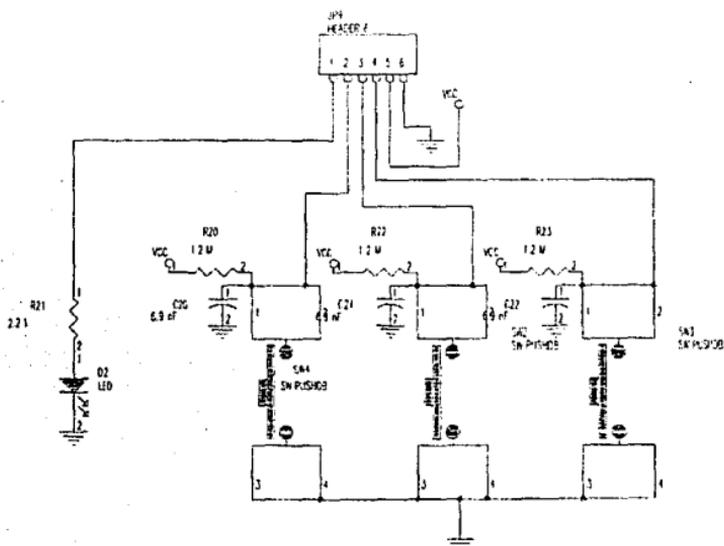
Para su manejo, a este teclado se le asignó el uso de una de las líneas de interrupción externa del microcontrolador. Simultáneamente, (ver fig.#32) 3 líneas del puerto P1 del microcontrolador muestran el estado de cada una de las teclas; así el sistema determina cual de las 3 teclas fue accionada, revisando el estado de cada línea. El manejo del teclado puede ser hecho por dos métodos: Atención a interrupciones o por revisión del estado (poleo) de las señales del puerto.

Por otra parte, los botones del ratón fueron conectados a la segunda de las interrupciones externas y a otras 2 líneas del puerto

P1 para dar entrada a los comandos de medición. De igual manera, la interrupción se genera cuando cualquiera de las teclas es accionada. La señal es transmitida al procesador el cual revisa el estado de las líneas del puerto para saber cual tecla fue operada.

Ambos circuitos cuentan con filtros RC pasa bajas para reducir el ruido en las señales. En el caso del teclado de comandos el filtro tiene una frecuencia de corte de 120 Hz. En el caso del teclado de medición la frecuencia de corte es de 6 KHz.

Fig.#32 Teclado de operación con el sistema.

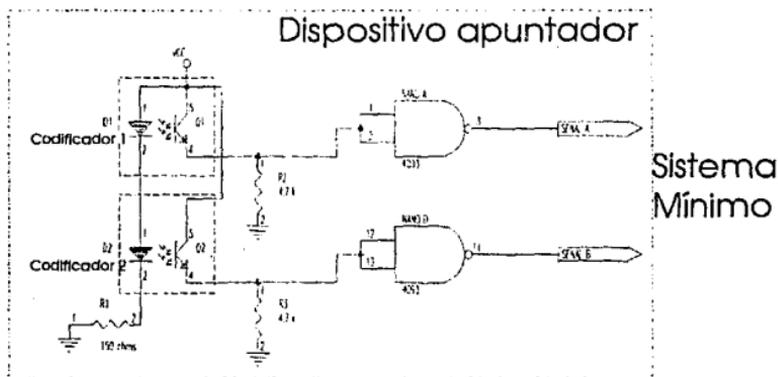


4.2.2.2. Interface con el sensor.

Esta etapa tiene como objetivo el transformar, para su conteo, las señales que entrega en pulsos el dispositivo apuntador. Parte de estas funciones son realizadas dentro de la electrónica del ratón, por lo que primero explicaremos las operaciones que realiza el ratón seleccionado y a partir de estas, la forma en que implementamos la interface.

El codificador óptico con el que cuenta el ratón esta formado por dos LED's, dos fototransistores, el disco codificador y una retícula que limita la cantidad de luz que recibe el sensor. El fototransistor esta configurado como emisor común, polarizado con 5 Volts y con una resistencia de 4.7 k Ω que lo conecta a tierra, como se muestra en la figura #33.

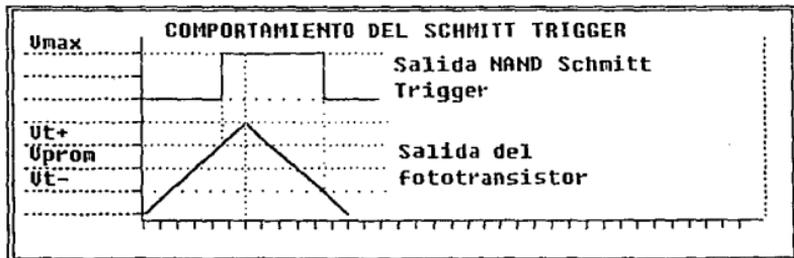
Fig.#33 Esquema eléctrico del dispositivo apuntador



El emisor está conectado a una compuerta NAND de entrada SCHMITT TRIGGER, configurada como inversor, con el fin de evitar los efectos del ruido. La compuerta no se dispara en el nivel medio de la transición, como lo realizan los TTL comúnmente; en su lugar, el

nivel de voltaje al que se dispara depende del tipo de transición que se realiza; de ser positiva el voltaje de disparo esta por encima del valor medio de la transición. En caso contrario, el voltaje de disparo esta por debajo del nivel medio de la transición. De esta forma, se reduce la posibilidad de que el ruido genere disparos falsos. La diferencia entre el valor de voltaje de ambos niveles se le conoce como voltaje de histéresis [25]. En el caso del circuito 4093 este valor es típicamente de 1.5 V. La función de transferencia de la figura#34, nos muestra el comportamiento del circuito.

Fig.#34 Disparo del circuito monoestable



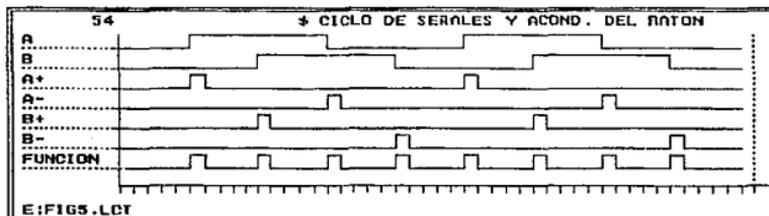
El uso de la compuerta SCHMITT TRIGGER nos permite darle a la señal forma de pulsos, eliminando el ruido y los periodos largos de transición de la señal. Este circuito en lugar de activarse por el cruce de un nivel de voltaje, espera a que la señal se aproxime al valor máximo si se encuentra en estado bajo o viceversa. Si el ruido de una señal no alcanza el nivel de transición este no cambiará. Esto se puede ver en la fig. #34. Esta es una forma de minimizar el efecto del ruido o alteraciones en la señal.

A partir de estas señales que entrega el sensor, se observa que los flancos de subida de los pulsos que entrega la NAND representan los máximos de la señal, que se suceden cuando esta pasando luz a través del decodificador. Mientras que los flancos de bajada representan los mínimos de la señal, cuando se ha cortado el paso a la luz. Como las secciones transparentes y obscuras son del mismo

tamaño, cada transición entre ellas debe ser considerada como un incremento igual de la distancia recorrida por el dispositivo apuntador. O sea, que la distancia que se tiene que recorrer para pasar de una sección oscura a una transparente, es la misma distancia que se recorre al pasar de una sección transparente a una oscura.

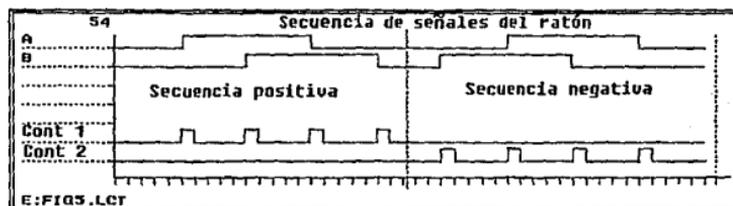
Las señales obtenidas de los dispositivos NAND de cada canal (mostradas en la fig. #35, como A+ y B+, para las transiciones positivas de cada canal, A- B- para los negativas) son sumadas y así obtenemos el desplazamiento del dispositivo.

Fig.#35 Obtención de los pulsos a contar.



Como fue mencionado al describir los codificadores ópticos, generalmente se utilizan dos juegos de detectores alineados con un defasamiento de la mitad de una sección transparente con el fin de detectar el sentido del movimiento, así como aumentar la resolución del dispositivo al doble. Por esta razón, además de determinar el momento de las transiciones al mover el dispositivo, debemos precisar el sentido del movimiento, tomando en cuenta el defasamiento de las señales. Por lo que, la suma no puede ser hecha directamente, sin determinar previamente el sentido del movimiento. Como se observa en la fig. #36, el sentido es función del estado en el que se encuentra una de las señales cuando la otra tiene una transición.

Fig.#36 Determinación del sentido del movimiento a partir de los defasamientos



Las señales son interpretadas de la forma siguiente:

| CANAL A | CANAL B | SENTIDO DE MOVIMIENTO |
|------------|---------|-----------------------|
| 0 → 1 (a↑) | 0 | Positivo |
| 0 → 1 (a↑) | 1 | Negativo |
| 1 → 0 (a↓) | 0 | Negativo |
| 1 → 0 (a↓) | 1 | Positivo |

De esta misma forma las transiciones en el canal B pueden ser usadas para determinar el sentido del movimiento.

| CANAL B | CANAL A | SENTIDO DE MOVIMIENTO |
|------------|---------|-----------------------|
| 0 → 1 (b↑) | 0 | Negativo |
| 0 → 1 (b↑) | 1 | Positivo |
| 1 → 0 (b↓) | 0 | Positivo |
| 1 → 0 (b↓) | 1 | Negativo |

De aquí se genera una tabla de verdad, para determinar las funciones a implementar:

| A↑ | A↓ | B↑ | B↓ | A | B | C. + | C. - |
|----|----|----|----|---|---|------|------|
| 1 | * | * | * | * | 0 | 1 | 0 |
| 1 | * | * | * | * | 1 | 0 | 1 |
| * | 1 | * | * | * | 0 | 0 | 1 |
| * | 1 | * | * | * | 1 | 1 | 0 |
| * | * | 1 | * | 0 | * | 0 | 1 |
| * | * | 1 | * | 1 | * | 1 | 0 |
| * | * | * | 1 | 0 | * | 1 | 0 |
| * | * | * | 1 | 1 | * | 0 | 1 |

* = No importa

1 = Activa

0 = No activa

Reduciendo esta tabla podemos determinar las funciones a implementar en nuestra lógica.

Una vez obtenidas las transiciones, es necesario implementar las funciones:

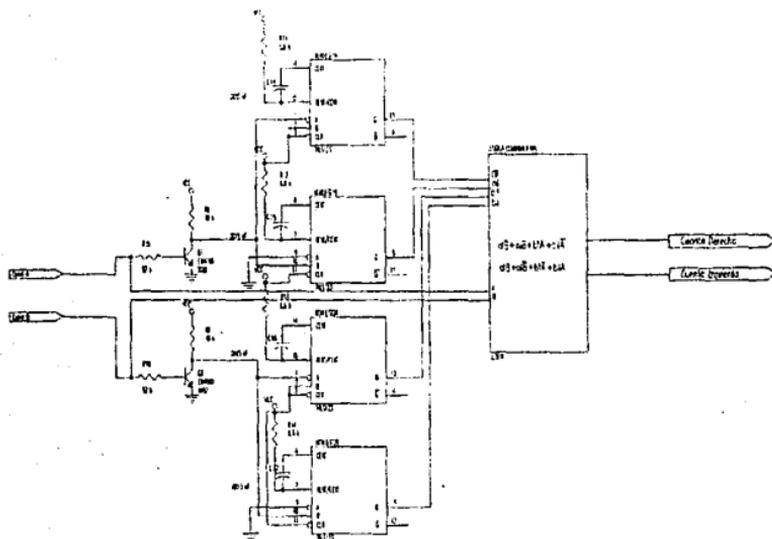
$$\text{Cuenta positiva} = a\uparrow\bar{B} + a\downarrow B + b\uparrow A + b\downarrow\bar{A}$$

$$\text{Cuenta negativa} = a\uparrow B + a\downarrow\bar{B} + b\uparrow\bar{A} + b\downarrow A$$

Esta función fue implantada en un circuito PAL (20L8), junto con las señales de los teclados.

Para detectar las transiciones usamos monoestables, los cuales se dispararán dependiendo del tipo de transición requerida. La figura #37 muestra la forma en que los monoestables se configuraron para detectar las transiciones.

Fig.#37 Diseño de enlace entre el ratón y el sistema de procesamiento.



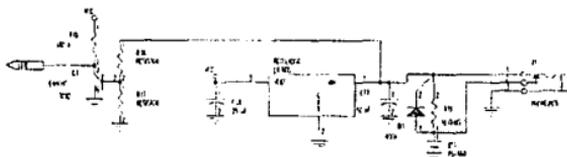
4.2.2.3. Fuente de poder.

La alimentación de energía se realiza en forma primaria por una pila electrolítica recargable PS 660 Power Sonic. Esta puede mantener 6 V con una capacidad de 6.5 A·hrs en 20 horas. El sistema cuenta además con un regulador integrado de 5 V a 1 A, tipo 7805 de

Motorola y un disipador de calor. Se dispone también de un enchufe para un eliminador externo de batería, con el cual es posible cargar a la pila.

Para recargar la batería se diseñó un circuito que limita el sentido de la corriente y que permite la recarga de la batería con seguridad (Fig. #38).

Fig.#38 Fuente de alimentación



Toda pila recargable tiene una corriente máxima a la que puede ser recargada, si se excede este valor se corre el peligro de hacer estallar la pila. Esta corriente es generalmente la máxima que se puede drenar de la pila; por tanto, es conveniente hacer siempre la recarga usando una fracción de dicha corriente. Debido a esto se limita la corriente por medio de una resistencia, de tal forma que nunca se sobrepase dicho limite. Los parámetros que tenemos que tomar en cuenta al realizar la recarga son: la diferencia de voltaje máximo en la recarga, la corriente de recarga y el tiempo en que se mantendrá el circuito funcionando. A partir del voltaje máximo y de la corriente de recarga obtenemos la resistencia que limita al circuito. La corriente de carga es múltiplo de la capacidad de la pila y del tiempo de recarga. En este caso, si la recarga se realiza con una fuente de 9 Vcd y se pretende recargar la batería en aprox. 10 hrs. tenemos:

$$R_{carga} = \left(\frac{V_{lim} - V_{pila}}{C \div T} \right) = \left(\frac{9 - 5}{6.5 \div 10} \right) = 6.1538 \Omega$$

La disipación de energía que tiene que soportar la resistencia de carga esta dada por la fórmula:

$$W_r = (V_{lim} - V_{bat}) * (I_{carga}) = (9 - 5) \times .65 = 2.6 W$$

Es necesario usar entonces una resistencia de 6.2Ω y de 2.6 W de capacidad de disipación.

V. DESCRIPCIÓN OPERATIVA DEL SISTEMA.

INTRODUCCIÓN.

El objetivo de hacer un sistema mínimo de desarrollo es tener la capacidad de solucionar distintos tipos de problemas por medio de una programación específica (software) y no la implantación física específica (hardware). Es por esto que la mayor parte de la solución del problema de medición esta realizado principalmente en forma de programa.

En este capítulo primero definimos la forma en que fue concebida la operación del sistema de medición: como se tomarían y almacenarían las medidas, su manipulación, etc.

A continuación, se describen las distintas rutinas que fueron implementadas para su operación: los procedimientos de adquisición y conversión de datos, los de comunicaciones, y finalmente, se describen las rutinas específicas para la edición y manipulación de la información.

Como antecedente a las rutinas se ha incluido en el apéndice una descripción de los modos de direccionamiento del microcontrolador, y las rutinas de uso genérico, que son empleadas en varios procedimientos y que pueden ser usadas fácilmente en otras aplicaciones

5.1. El Ictiómetro Automático.

El Ictiómetro Automático (como se le denomino a la aplicación del sistema diseñado) es un instrumento capaz de medir la longitud de un pez y almacenar este dato para su posterior procesamiento. El sistema fue concebido para ejecutar 3 tareas básicas:

-
- Medición y almacenamiento de datos.
 - Revisión y edición de datos.
 - Transferencia de datos a una Computadora.

Cada una de estas tareas es accesada por medio del programa de menú, que presenta las distintas opciones y ejecuta las rutinas correspondientes. Este programa arranca en el momento de iniciar el sistema, haciendo uso del teclado del Ictiómetro para recibir los comandos de su operación.

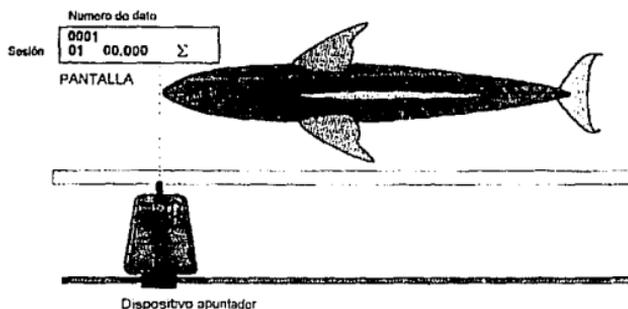
El equipo de medición esta formado por una unidad de procesamiento, y una regla de medición. En la unidad de procesamiento se encuentra el sistema mínimo, más la interface con el dispositivo apuntador; este último se encuentra montado en la regla de medición.

5.1.1. Medición y almacenamiento de datos.

Las tareas de medición y almacenamiento de datos se realizan con un esquema de trabajo basado en sesiones. Cada vez que es accesada la rutina se inicia una nueva sesión. Los datos que son capturados ocupan un espacio de memoria; cuando se terminan de tomar las medidas se cierra la sesión. Con el uso de este esquema de trabajo se facilita la organización de los datos; ya que cada sesión puede contener los especímenes de una cierta clase, o de una zona. Antes de iniciar la tarea de medir y almacenar longitudes, es necesario indicar que tipo de medidas se tomarán durante la sesión: longitudes, anchos o ambas.

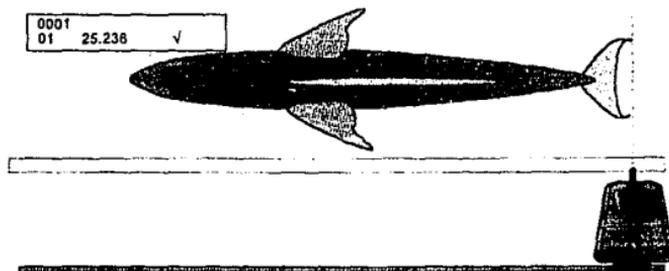
Para la medición se utiliza al ratón como sistema de entrada de datos. El sistema se conecta a una regla de medición, en la cual esta montado el dispositivo apuntador; el pescador se coloca frente a él y se toma la medida. Al comenzar una medición, se alinea primero el ratón a la izquierda en el inicio del pescador. Al oprimir el botón izquierdo; principia el conteo de distancia a partir de esa posición (fig. #39).

Fig.#39 Alineamiento e inicio de la lectura



Soltando el botón izquierdo, se recorre el dispositivo hasta alinearlo con el extremo a la derecha del pescado y se presiona una vez el botón derecho (fig. #40) para finalizar la medición.. Cuando esto sucede se detiene el conteo y se almacena el resultado. Se procede de esta forma con los demás especímenes.

Fig.#40 Alineamiento final y captura del dato



La pantalla de cristal liquido presenta en todo momento el valor actual de la distancia recorrida, así como el número de medición y el número de sesión correspondiente. Adicionalmente, cuando es oprimido el botón de inicio de lectura se despliega en la pantalla un

signo de Σ (F6H), para indicar que se están tomando la medida; el signo de \surd (E8H) indica cuando la medición se ha detenido y el dato ha sido almacenado exitosamente. Las medidas se despliegan en pulgadas, ya que es directo su manejo porque la resolución del dispositivo apuntador es una fracción de pulgadas.

5.1.2. Revisión y edición de datos.

En este procedimiento, el Ictiómetro permite la manipulación de los datos almacenados. El sistema consta de 3 formas de edición de los datos: borrado de todos los datos en la memoria, borrado de una sesión y borrado o edición de un dato. Los datos almacenados en la memoria del Ictiómetro pierden su utilidad en el momento en que se ha realizado su transmisión a la computadora. Sin embargo no es conveniente que automáticamente se borren, puesto que existe la posibilidad de que los datos puedan ser necesitados nuevamente. La rutina de borrado de todos los datos de la memoria se diseñó para ser autónoma, y se realiza por la petición específica del operador y requiere de confirmación.

La rutina de borrado de una sesión permite el eliminar el archivo de una sesión sin afectar el resto de los datos. Al comenzar la rutina se pregunta cuál de las sesiones es la que se debe de eliminar, se pide una confirmación y se realiza la operación.

La rutina de borrado o edición de un dato permite eliminar o corregir datos que han sido mal tomados. Primero la rutina pide que se escoja la sesión en la que fue tomado el dato, luego se procede a escoger el dato erróneo. Una vez escogido pregunta que acción se piensa realizar: borrado, o corrección de datos. Cuando se trata de borrado de datos, se pide una confirmación y se realiza el borrado; si se desea corregir el dato se ejecuta de nuevo la rutina de medición.

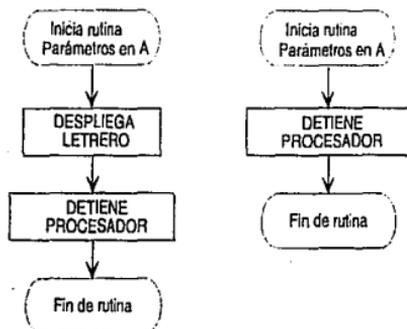
5.1.3. Transferencia de datos a una Computadora

El procedimiento de comunicación se encarga de transmitir los datos a una computadora para su procesamiento. La comunicación es automática, por lo que tan solo es necesario indicar al sistema a que velocidad se transmitirán los datos (baudaje), por parte del usuario solo requiere de conocimientos básicos del uso de una computadora. El protocolo de comunicación y la manipulación de posibles errores en la comunicación se realiza en la computadora, por medio de un programa que se comunica con el Ictiómetro.

5.1.4. Menús de opciones.

Para facilitarle al usuario la operación del equipo, se implementaron dos métodos diferentes para el despliegue de opciones, de tal forma que sea sencillo interactuar con él. El primero despliega un letrero ya almacenado en memoria y detiene la operación del microcontrolador (Fig. #41), hasta que es accionada alguna tecla. Este tipo de menú de opciones siempre presenta la misma información, por lo que también se hace cargo de manejar el despliegue. El segundo no despliega ningún letrero (este debe ser previamente desplegado) por lo que tan solo esta encargado de la detención del procesador. Además este último tiene la ventaja de poder desplegar del letrero cualquier otro dato variable, que sirva de guía al usuario.

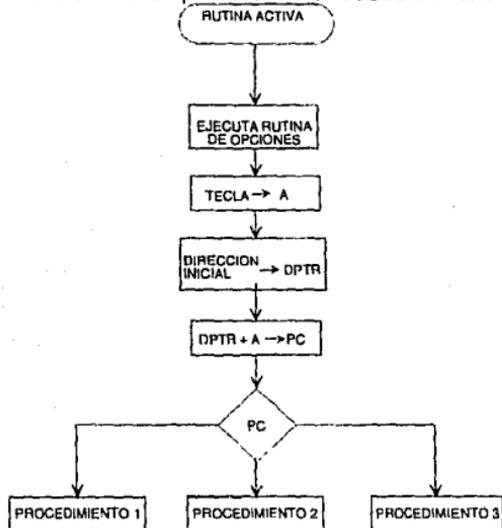
Fig.#41 Paro del procesador en las rutinas de despliegue.



El teclado genera una interrupción cuando es accionado, reactivando al procesador. Cuando esto ocurre la rutina de atención a la interrupción lee el puerto para determinar que tecla a sido oprimida. Esta rutina también verifica que la tecla haya sido soltada antes de permitir que continúe la operación.

Las rutinas no realizan ningún trabajo de procesamiento sobre la información de la tecla accionada, tan solo sirven para llevar el control de la operación de paro de procesos. El procesamiento de los datos recibidos para elegir las acciones adecuadas, es realizado por las propias rutinas que presentaron las opciones; éstas generalmente son llevadas a cabo con un salto tipo "en caso de" (Fig. #42). Para esto, es asignado un valor numérico a una variable dependiendo de la tecla accionada; luego, esta variable sirve como índice de la línea de programa donde se encuentra el salto a la rutina adecuada. Aunque este método esta limitado a 3 opciones, ya que el teclado es de 3 funciones, puede ser fácilmente generalizado a más; evitando líneas de código repetitivas, como en el caso de los saltos con instrucciones tipo "Si... Entonces".

Fig.#42 Elección de procedimiento desde el menú principal



Adicionalmente, se creó una rutina para desplegar letreros por un lapso de tiempo, con el fin de mostrar información o instrucciones. Esta rutina ejecuta las rutinas de despliegue de información y se mantiene haciendo un ciclo, cuando acaba, regresa el control a la rutina que la llamó.

5.1.5. Verificación del perro guardián.

Para que el dispositivo de verificación no genere una señal de "reinicio de sistema" es necesario, cada cierto tiempo, enviar una señal que confirme la buena operación del aparato. Después de un periodo de 0.6s (llamado Tmd) y si no se ha recibido la señal de verificación por parte del microcontrolador el dispositivo de verificación activa la señal de reinicio (reset). Por este motivo, es necesario que todas las rutinas generen periódicamente dicha señal.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

La señal se encuentra conectada a la terminal 6 del puerto 1 (P1.6) y es activada por los programas a través de una llamada a una subrutina, la cual se encarga de cambiar el estado presente en la terminal. Este cambio de estado es mantenido durante el lapso de ejecución de dos instrucciones "no operación"; después es restablecido su valor original. En total, el tiempo que transcurre activada la señal es de 36 ciclos de reloj (3 μ s), que es un lapso mucho más largo que el mínimo requerido para que funcione (40 ns).

La señal es activada por la transición negativa, por lo que es importante restablecer siempre el estado alto. De esta forma, cuando se realiza el primer cambio de estado, se realiza la verificación. Al usar la función de complementación garantizamos que siempre se va a generar la señal de verificación (transición negativa) ya sea en la primera o en la segunda, sin importar el estado que guardaba el puerto en ese momento (Fig. #43).

Fig.#43 Temporización de las señales en el "Perro guardián"



Como se mencionó en el capítulo II, cuando nos disponemos a detener la operación del microcontrolador 80C31 es necesario detener también la operación del perro guardián; para esto usamos la línea 5 del puerto 1 (P1.5). Cuando se le asigna un nivel bajo se acciona la función de borrado del monoestable sin que se generen

pulsos de disparo (característica del 74LS423); por lo que se desactiva hasta que es vuelto a disparar por el microcontrolador, una vez que se ha reiniciado la operación del sistema. Cabe destacar la importancia de esta reactivación, pues si esta no se ejecuta, se corre el peligro de que alguna falla cause graves daños a la información que guarda el microcontrolador.

5.1.6. Estados de ahorro de energía.

Cada vez que se realiza el método de menú de opciones, la operación del microcontrolador es detenida con el objeto de reducir el consumo de energía. Cuando el menú es presentado en la pantalla, se realiza la detención del procesador. Esto se logra escribiendo un valor de 1 en el bit 0 del registro de control de energía (PCON). Si esto se realiza directamente, al cabo de medio segundo se generará una señal de reinicio (RESET) por el perro guardián. Para evitar este "reset", es necesario que se detenga la operación del perro guardián. El bit 5 del puerto 1 controla la señal de borrado, por lo que antes de realizar la detención del microcontrolador se genera una transición negativa en esta terminal. Para asegurar que dicha transición se va a realizar sin importar el valor que actualmente contenga la terminal del puerto, se realizan 2 instrucciones de complemento de bit. De esta forma, en cualquiera de las dos se realiza la transición deseada; deteniendo la operación del perro guardián. Cuando se ha realizado la detención es necesario que se genere algún tipo de interrupción para salir de ella. Como se explico para la rutina de "menú de opciones" las interrupciones se generan en el momento de accionar una tecla, ejecutándose inmediatamente una rutina de atención al teclado.

5.2. Procedimiento de adquisición y conversión de datos.

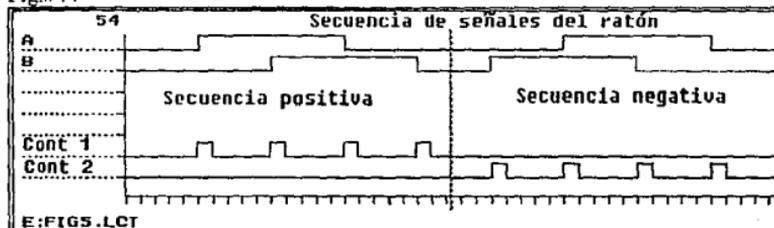
Interpretación de las señales.

Como se explico anteriormente, las señales que envía el dispositivo apuntador son pulsos que indican el incremento de su

posición en un sentido. Cada pulso, que lo definiremos como un paso, representa un desplazamiento de 1/200 de pulgada. Aunque el ratón detecta el movimiento en 2 ejes, la medición solo requiere de uno, por lo que se fija en una guía, limitando su movimiento a un solo eje (X).

En el eje existen dos canales: A y B con un defasamiento de 90 grados para determinar el sentido y para duplicar la resolución; Definimos como sentido positivo a los movimientos hacia la derecha del "ratón" (cuando el canal A esta adelante del canal B) y negativos a la izquierda (cuando el canal A esta detrás del canal B). El desplazamiento total es la diferencia entre el total de pasos hacia la izquierda y la derecha (fig. #44).

Fig.#44



5.2.1. Conteo de Pulsos.

El conteo de pulsos es manejado directamente por los contadores internos al ratón. Cada paso representa 1/200 de pulgada; o sea, 0.005 pulgadas. Una pulgada puede ser representada en un byte de información, con el número de pasos. Por esto, los contadores se configuran para trabajar como contadores de 8 bits con recarga de un valor inicial. Este valor inicial es de 56 (38H); a partir de él se cuentan 200 pasos y se realiza la recarga para volver a iniciar el conteo. Cada vez que se ha completado la cuenta y se sucede una

recarga, se genera una interrupción. La rutina de atención se encarga de incrementar un registro que mantiene el número de pulgadas que se han recorrido en algún sentido.

A pesar de que, el desplazamiento total es la diferencia entre los desplazamiento de cada sentido, la diferencia no se obtiene restando directamente los 4 registros, de pulgada y de fracciones de pulgadas, ya que se debe de tomar en cuenta que:

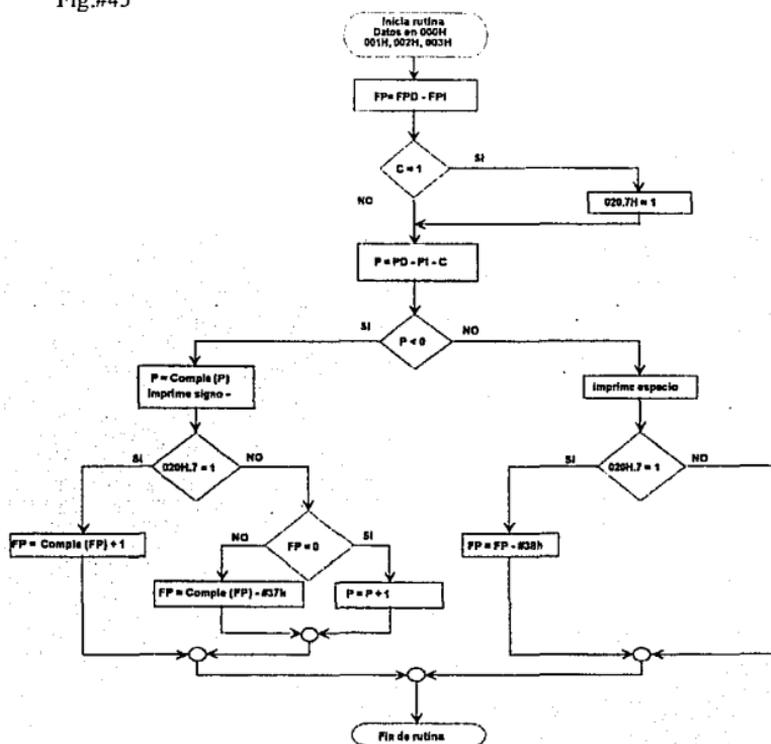
1. Si la diferencia entre las fracciones es negativa, entonces es necesario decrementar en una unidad el valor de las pulgadas enteras, cuando la suma de los enteros es superior a 1. Las fracciones serán el complemento a 200 de la diferencia.
2. Análogamente, cuando la diferencia entre las fracciones es positiva es necesario incrementar una unidad el valor de las pulgadas enteras cuando su suma es menor de -1. Las fracciones serán el complemento a -200 de la diferencia.
3. El registro que contiene el número de fracciones de pulgadas recorridas se inicia con un valor de 56, por lo que es necesario eliminarlo de los cálculos.
4. Cuando la diferencia de las pulgadas enteras es igual a cero, el signo lo determinará el resultado de la diferencia entre las fracciones.

La rutina de medición configura primeramente la operación de contadores, inicia otros registros y activa la interrupción que accesa a los botones del ratón. Esta rutina se encarga de:

1. Desplegar el letrero de medición en la pantalla .
2. Desplegar variables como el número de dato y de archivo.
3. Ejecutar la subrutina de conteo y conversión.
4. Desplegar resultados.
5. Atender al teclado.

La subrutina de conteo y conversión (ver fig. #45) es la encargada de realizar las operaciones necesarias para la determinación de la distancia absoluta recorrida. Esta rutina también evalúa las condiciones para determinar el magnitud y sentido del movimiento, que citamos con anterioridad. A continuación se describe lógicamente el proceso.

Fig.#45



Como se dijo antes, se realiza un conteo separado de las pulgadas enteras y de las fracciones recorridas en cada sentido. Primero efectúa la resta del valor de las fracciones de pulgada y se determina si es que se trata de un número negativo; de serlo, se indica el resultado en un registro que es usado como bandera (20H). Después, se hace, la resta de la parte entera tomando en cuenta el acarreo de la operación anterior.

Si el resultado es negativo, se procede a enviar el signo negativo; se verifica el signo de la diferencia de las fracciones. Cuando las fracciones son negativas, se procede a hacer el complemento a 2 y se almacena el dato. Cuando se hace el complemento a 2 de un byte, se determina el número de unidades que le falta a ese número para ser 256. Debido a que el conteo no se realiza hasta 256, sino hasta 200, el resultado es entonces mayor en 56 (37H) unidades; por lo que deben serle restadas para ajustar el resultado. Si las fracciones son un número positivo, se verifica si se trata de cero; ya que en este caso se debe de incrementar en 1 el valor de los enteros. De no serlo, se procede a complementar el valor de las fracciones.

Cuando el resultado de la diferencia de los enteros es positiva, se envía un espacio y se determina el signo de las fracciones. Cuando la diferencia de las fracciones es positiva se termina el proceso. De lo contrario, cuando es negativa, se realiza el complemento a 2 y se substraen 37H, para ajustar el complemento a 2. La rutina termina restableciendo las variables del sistema.

5.2.2. Estructura de almacenamiento.

Como se trata de una memoria de 64 Kbits, (8k x 8), se cuenta con una capacidad de almacenamiento total de 8,192 bytes. Cada medida ocupa un espacio de 2 bytes, por lo que la memoria sólo puede albergar 4,097 datos. Además, era necesario almacenar algunos otros datos; por lo que se decidió limitar la capacidad de

almacenamiento de datos a 4000 y dejar los 192 bytes restantes para datos de control o de sistema.

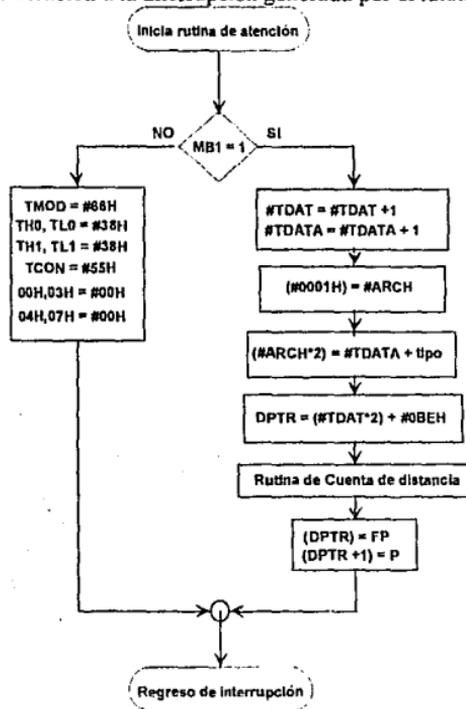
Además de los datos, se almacena también el número total de datos por archivo que se han grabado y se incluye un registro que indica si la memoria tiene datos o si esta completamente limpia.

Los datos en la memoria se organizan en archivos. Cada vez que se inicia una sesión se inicia un nuevo archivo, de esta forma se puede mantener un orden de como fueron tomados los datos. Puede haber hasta 94 archivos en la memoria. Cada archivo esta representado por 2 registros que guardan la cantidad de datos que le corresponden y el tipo de medidas que le fueron tomadas a los especímenes durante la sesión: largos, anchos o largos y anchos. La indicación del tipo de archivo se encuentra en los bits 5 y 6 del segundo byte del archivo. A partir de la línea 00C0H, se encuentran todos los datos en forma continúa; de esta forma se aprovecha al máximo el espacio.

5.2.3. Operación del ratón.

Los botones del "ratón" están encargados de iniciar y terminar el conteo. Cuando alguna de las teclas ha sido accionada, el microcontrolador recibe la señal en forma de una interrupción externa (fig. #46). La rutina de atención determina, antes que otra cosa, cual tecla ha sido oprimida y salta al procedimiento adecuado. Si se ha tocado la tecla izquierda, es señal de que se quiere reiniciar la operación de conteo; por lo que se borra el contenido de todas las variables involucradas y se procede a activar la operación de los contadores, escribiendo en el bit de operación de contadores de su registro de control. Esta es siempre la primera operación, antes de tomar una medida.

Fig.#46 Rutina de atención a la interrupción generada por el ratón



El segundo botón es el encargado de dar por terminado el conteo del recorrido y de almacenar el dato en memoria.

5.2.4. Almacenamiento de datos.

Cuando se inicia la rutina de medición, se revisa el contenido de la memoria para saber si ya se han almacenado datos, cuantos archivos se han usado y la cantidad de datos que hay en cada uno. Se inicia un contador que mantiene la cantidad de datos totales en la

memoria y otro para el número del archivo correspondiente, se asigna el número de archivo que le corresponde a la sesión actual a un registro interno y se procede a realizar la medición. Cuando se ha accionado la tecla derecha del ratón, se detiene la medición y se salva el dato. Primero se guarda la indicación de memoria usada, 00H, en la dirección 0000H. Se calcula el total de archivos alcanzado hasta ese momento en la dirección 0001H y se determina la dirección que le corresponde a los datos del encabezado del archivo (dependiendo del número de archivo que se trate), haciendo un corrimiento de este número. Luego se almacena la información del archivo y se procede al cálculo de la dirección donde almacenar el dato (también con un corrimiento) agregando un valor de referencia, en este caso BFH. Finalmente se ejecuta por última vez la rutina de conteo y se almacenan los resultados. En la figura #47 se describe la organización de la memoria.

Cada vez que se escribe a la memoria EEPROM, se tiene que esperar la señal de terminación del ciclo de escritura. Cuando se ha terminado, se procede a restablecer la condiciones del sistema y en caso de ser necesario, a desplegar otro letrero.

Fig.#47 Mapa de memoria de datos

| | |
|-------|---|
| 0000H | MEMORIA EN USO = 0000H |
| 0001H | NUM. DE ARCHIVOS = #ARCH |
| 0002H | NUM. DE DATOS DEL ARCHIVO 1 (LSB) = #TDATAL |
| 0003H | NUM. DE DATOS DEL ARCHIVO 1 (MSB) = #DATAH |
| | |
| 00BEH | NUM. DE DATOS DEL ARCHIVO 92 (LSB) = #TDATAL |
| 00BFH | NUM. DE DATOS DEL ARCHIVO 92 (MSB) = #DATAH |
| 00C0H | DATO 1 ARCH 1 (FRACCIONES DE PULGADA) |
| 00C1H | DATO 1 ARCH 1 (ENTEROS DE PULGADA) |
| | |
| 0FFFH | DATO n ARCH m (ENTEROS DE PULGADA) |

5.3. Procedimiento de comunicaciones.

Enlace del sistema.

Como se ha dicho, el enlace con la computadora se realizó a través del puerto serial; por medio de un cable tipo "null modem" de 3 hilos; la comunicación se realizó de forma asíncrona. La detección de cada dato es realizada con la ayuda de un primer pulso que indica cuando se inicia la transmisión del dato. Este bit, que es cero, es llamado "bit de inicio". Luego se procede a detectar el carácter transmitido y por último se espera un bit de paro. La transmisión, por tanto, será de 10 bits y sin ningún bit para probar la paridad.

El sistema permite satisfacer distintos requerimientos de comunicación, con el objeto de enlazarse a diferentes tipos de PC's. Para esto, el programa permite cambiar la velocidad de transmisión de los datos. Se manejan actualmente 3 velocidades: 1200, 2400 y 9600 bps. Para alcanzar dichas velocidades fue necesario programar el temporizador de acuerdo a la siguiente tabla:

| TASA DE BAUDS | SMOD | MODO | VALOR DE RECARGA (TH1) |
|---------------|------|------|------------------------|
| 9600 | 0 | 2 | FDh |
| 2400 | 0 | 2 | F4h |
| 1200 | 0 | 2 | E8h |

De acuerdo a la fórmula de la tasa de bauds:

$$\text{Tasa de Bauds} = \frac{2 \text{ SMOD}}{32} \times \left(\frac{\text{Frecuencia de oscilación}}{12 \times (256 - (\text{TH1}))} \right)$$

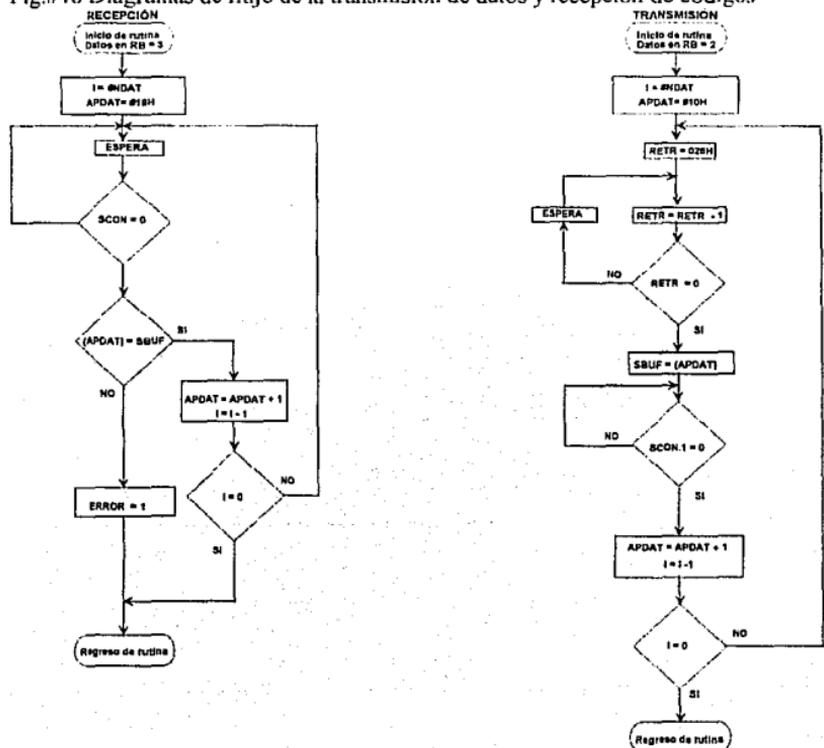
Donde:

SMOD = Es el valor del bit 7 del registro PCON, con el cual se indica el valor del preescalamiento de la señal de reloj.

TH1 = Valor de recarga del registro, a partir del que se inicia la cuenta.

La comunicación es realizada por 2 rutinas: transmisión y recepción (cuyos diagramas de flujo están en la fig. #48). El microcontrolador se encarga únicamente de la comunicación, espera a que se haya realizado toda la transmisión o a que todos los datos hayan sido recibidos; por lo que no es necesario el uso de las interrupciones para controlar el proceso. Cada una de las subrutinas de recepción y transmisión usan un bloque de registros internos del 8031 como buffer. De esta forma la rutina escribe en el buffer los

Fig.#48 Diagramas de flujo de la transmisión de datos y recepción de códigos



datos a transmitir, o los que espera recibir, e indica en un registro su cantidad. Hasta 8 datos pueden ser transmitidos de esta forma. La rutina de transmisión envía uno a uno los datos, cuando el buffer de transmisión se ha vaciado, se carga un nuevo dato. Este proceso continua hasta que se han enviado los datos. Ya que la PC solo envía códigos que indican si puede o no continuar la comunicación, en la recepción se espera a que el mensaje haya llegado y se compara con lo que se esperaba; si concuerdan, prosigue la comunicación.

Protocolo de comunicación.

Para mantener un control en la comunicación se elaboró un protocolo. Esta es una forma sencilla de mantener un orden en la comunicación y nos sirve para detectar posibles errores. A partir de este protocolo se crearon los programas para la PC y el lctiómetro. Los pasos que se realizan para la lograr la transmisión de los datos son los siguientes:

INICIO DE COMUNICACIÓN

1. La PC transmite el código de inicio de comunicación, representado por 4 caracteres: NULL, NULL, P, C.
2. Cuando el lctiómetro recibe el inicio, este transmite su código de aceptación: NULL, NULL, 3, 1. Estos 2 códigos dan la personalidad del dispositivo.
3. Si la computadora ha recibido la aceptación, entonces se envía el código de confirmación: NULL, NULL, STX. Para este momento se ha confirmado que la línea se encuentra en buen estado ya que se han reconocido mutuamente los dispositivos.
4. Se procede a informar al lctiómetro de las capacidades de velocidad de comunicación de la PC. Se transmite un dato que informa al lctiómetro cuanto debe de esperar para que la PC pueda procesar el dato transmitido. Si el dato es cero entonces se procede a realizar la rutina de optimización para realizar la comunicación. Estos procedimientos se explicaran con más detalle en la siguiente sección.

ENVÍO DE ESTADÍSTICAS

5. Si la memoria contiene datos, se procede a realizar la transmisión. Se envía el número de archivos que contiene la memoria: NULL, NULL, # archivos. Si la PC ha recibido estos datos contesta con el código de confirmación. Si por alguna razón la memoria indica que no existen datos, se genera un error, realizándose otra rutina.
6. Se proceden a enviar los encabezados de los archivos: número de datos por archivo y tipo, empezando por el bit menos significativo. Cuando se ha completado el envío de todos los encabezados se espera el código de confirmación.
7. A modo de comprobación, se hace un cálculo del total de datos que contiene la memoria y se envían con el siguiente formato: NULL, NULL, <# total de datos>MSB, <# total de datos>LSB. Si el dato es correcto se espera el código de confirmación.

TRANSMISIÓN DE LA INFORMACIÓN

8. Se envían, como un solo bloque, todos los datos almacenados.
9. Se envía el código de fin de transmisión de bloque: NULL, NULL, EOT
10. El Ictiómetro recibe como comprobación el número total de datos recibidos por la PC. Si no corresponden se genera un error. Esto termina la comunicación.

Es importante mencionar algunas consideraciones:

- Los errores en la comunicación son evaluados después de una recepción, si los datos no concuerdan con el protocolo.
- Cuando se genera un error, tanto en la PC como en el Ictiómetro, siempre se regresa al inicio de la comunicación. De esta forma se hacen reintentos consecutivos.
- En caso de que por algún error la memoria indique que no hay datos, es posible realizar una operación de salvamento del contenido completo de la memoria. Cuando esto sucede, los datos (ya almacenados en la PC)

no tienen formato; por lo que es necesario realizar una edición manual en la computadora con algún editor de textos.

- Sólo se ha presentado el flujo de la comunicación, el programa es más amplio y realiza otras operaciones.

Transmisión de la información.

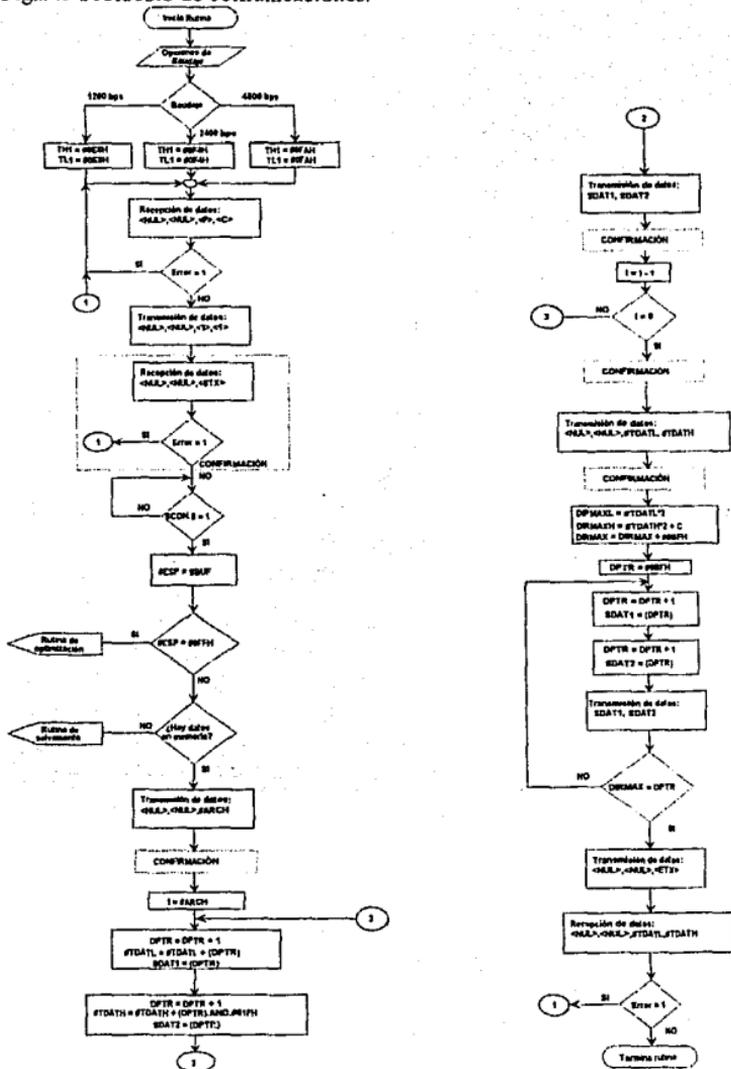
El programa de comunicaciones consta de 3 rutinas: descarga de datos, salvamento de memoria y optimización de comunicación (fig.#49). Estas rutinas se activan dependiendo de las circunstancias en las que se estén ejecutando. La selección del baudaje la tiene que se la misma en la PC y en el lctiómetro.

Cada vez que es iniciado el procedimiento de comunicación, es necesario indicar primero a que velocidad se va a llevar a cabo el enlace. Entonces se inicia la comunicación de acuerdo al protocolo; si se realiza el reconocimiento de máquinas, se procede a recibir el número de esperas, un valor entre 1 y 255. Se continúa con la rutina de descarga de datos.

La rutina de descarga de datos sigue paso a paso al protocolo de comunicación. Comprueba que la memoria tenga datos y captura el número de archivos usados; de no haberlos se genera un error y se procede con la rutina de salvamento. Si hay datos, se procede a la lectura y transmisión de los encabezados de cada archivo. Al mismo tiempo, se va haciendo la cuenta del total de datos que hay en memoria. Para esto, es necesario manipular los encabezados, haciendo uso sólo de los bits que representan la cantidad de datos y no el tipo de archivo de que se trate.

De acuerdo al total de datos a transmitir, se determina la dirección final de memoria en la que se encuentran los datos. Luego, se realiza un ciclo en el que se capturan los dos bytes que le corresponden a cada dato. Estos son transmitidos y cuando se alcanza la dirección final, se sale del ciclo.

Fig.#49 Protocolo de comunicaciones.



La rutina de salvamento de memoria se realiza cuando se pretende efectuar una descarga de datos, pero la memoria indica estar vacía. Esta condición puede suceder si por algún motivo se perdió la información del primer registro de la memoria. En este caso, se puede realizar un salvamento de datos descargando el contenido de toda la memoria en la PC. Para esto, se pide primero confirmar la operación, de aceptarse, los datos de la memoria son capturados y enviados inmediatamente. Esta operación se realiza sobre toda la memoria.

5.4. Procedimiento de manipulación y edición de la información.

5.4.1. Verificación del estado de la memoria.

Para corregir y borrar los datos erróneos, se creó el procedimiento de edición de información, que permite revisar y borrar datos o archivos. La rutina de edición presenta 3 opciones: Borrado de memoria, Borrado de archivo y Eliminación o edición de datos.

Cada vez que se accesa el procedimiento de edición, primero se verifica el estado de la memoria leyendo el contenido del registro de uso de memoria (0000H). Si este indica que hay datos continúa el proceso, de lo contrario se despliega un letrero de información y se regresa al menú inicial. De existir información se presenta un menú mostrando las posibles formas de edición.

5.4.2. Borrado de memoria.

Cada vez que se han descargado los datos en la PC, es importante limpiar el contenido de la memoria. Para esto se escribe en toda la memoria un valor FFH. De esta forma, si por alguna razón hay un error en el registro de uso de memoria y se realiza un salvamento de datos, es fácil reconocer hasta donde llegan los datos.

La rutina de borrado pide la confirmación antes de realizar la operación. Después realiza un ciclo iterativo, en el que se hace la escritura, se espera su verificación, se incrementa la dirección y se cierra el ciclo. Cuando se ha terminado se despliega un letrero y se regresa al menú inicial.

5.4.3. Borrado de archivo.

Esta rutina es más complicada; ya que, cuando se necesita borrar un archivo, es necesario recorrer los demás archivos para evitar que queden espacios vacíos. Se crearon 3 subrutinas para hacer el manejo de datos: Determinación de archivo, Determinación de direcciones y Borrar parte.

5.4.3.1. Determinación de archivo.

Esta rutina es la que lleva el control del proceso. Primero lee de la memoria de datos la cantidad de archivos existentes en la memoria; se inicia un contador de archivo en el registro (1AH) con el valor de 1; se despliega un menú de opciones y se espera. Las opciones del menú son: "más", "menos" y "borra".

Cuando se acciona la opción de "más", se salta a una rutina que incrementa el contador de archivo. Si se sobrepasa el número total de archivos en la memoria, se reinicia el conteo con el valor de 1. De forma análoga trabaja la función de "menos"; se decrementa el valor del contador de archivos y cuando este disminuye a menos de 1, entonces se toma el número total de archivos.

Una vez elegido el número de archivo a eliminar, se escoge la opción 3: "borrar". La rutina pide una confirmación de operación, que de no ser aceptada regresa al menú inicial. Si es aceptada, entonces se inicia el cálculo de direcciones que le corresponden a dicho archivo. Se revisa si se trata del último archivo almacenado en la memoria; de serlo el procedimiento es más sencillo, pues sólo es

necesario descontarlo del contador de archivos en memoria. Esto es posible ya que al ser el último, no es necesario el optimizar la memoria, puesto que la siguiente vez que se inicie un archivo, este escribirá sobre el contenido anterior.

Cuando el archivo no es el último, se procede a determinar las direcciones donde inicia este y el archivo siguiente, así como la dirección donde terminan los datos en la memoria. Para esto se crearon las rutinas de determinación de dirección. Estas rutinas requieren que se les entregue como parámetro el número de archivo que se desea investigar. Sus resultados son organizados en memoria, para posteriormente servir de datos a la rutina de borrado.

Cuando se tienen las direcciones se ejecuta la rutina de borrar parte. Esta rutina recorre el contenido de una locación de memoria a otra. De esta forma se llena el espacio que ocupa el archivo borrado con datos de los siguientes archivos. Posteriormente, se procede a realizar lo mismo con los encabezados de los archivos; se determinan las direcciones de los encabezados y se realiza el movimiento de la información.

Por último, se decrementa el número total de archivos y se almacena este dato en la memoria.

5.4.3.2. Determinación de dirección.

La determinación de la dirección en la que se encuentran los datos de un archivo se realiza por medio de dos rutinas: conteo de datos y conversión a direcciones. Aunque estas dos rutinas generalmente trabajan juntas, se mantuvieron separadas porque en algunos casos se requieren independientemente.

La subrutina de conteo de datos encuentra el número de datos que existen desde el primer archivo hasta un archivo determinado. Para lograrlo requiere se le dé, en el registro 008H, el número de archivo hasta el cual se va a hacer el conteo. Además, requiere que previamente se haya inicializado al registro apuntador de datos

(DPTR) para que a partir de esa dirección realice el conteo. Esto también permite que la rutina cuente el número de datos entre uno y otro archivo. Con estos datos y si el número de archivo es distinto a cero, se procede a realizar el conteo de número de datos. Se realiza un ciclo iterativo, en el que se captura el dato y se va sumando al contenido de un par de registros de almacenamiento previamente inicializados. Luego se lee la parte baja del número de datos de un archivo y se suma con el registro; se lee entonces la parte alta, se eliminan los bits con la información del tipo de archivo y se efectúa la suma con el segundo registro de almacenamiento, tomando en cuenta el acarreo de la suma anterior. El ciclo se repite hasta alcanzar el número de archivos solicitado.

Ya que se ha obtenido el número de datos del inicio hasta el archivo correspondiente, es fácil encontrar la dirección de este último dato. La subrutina de Conversión a direcciones es la responsable de determinar la dirección de un dato. El número del dato (del que se requiere determinar su dirección) se debe encontrar en los registros 01BH (LSB) y en el 01CH (MSB). La dirección de un dato esta dada por su número y es necesario aplicarle las siguientes operaciones:

- Un corrimiento de un lugar a la izquierda; lo cuál representa una multiplicación por dos, ya que a cada dato lo componen 2 localidades de memoria.
- Se le agrega un valor de referencia, (offset), ya que los datos no empiezan en la dirección 0000H, sino en la 00C0H.

Al finalizar, los resultados son almacenados directamente en el registro apuntador de datos.

5.4.3.3. Borrar parte de datos

La reorganización de los datos la realiza una rutina denominada "borrar parte". Esta rutina recorre datos a las localidades que ocupan los datos que se desean eliminar. El programa utiliza como

parámetros la dirección donde se sobrescribirá y la dirección a partir de la cual están los datos que escribir. Estas direcciones deben de encontrarse en los registros 010H y 011H, y en el apuntador de datos (DPTR) respectivamente.

La rutina inicia con el desplegado de un letrero que indica la operación de optimización. Se inicia un ciclo incrementando el contenido del DPTR. Entonces se lee el primer dato que va relocalizar y guarda la dirección del DPTR en la pila de datos (STACK). Luego se carga en el DPTR el contenido de los registros de la dirección donde reescribir y se envía el dato a la memoria. Es necesario hacer una espera mientras la memoria almacena el nuevo dato. A continuación se incrementa la dirección en el DPTR y se guarda. Por último, se recupera de la pila de datos la localidad del nuevo dato a relocalizar y se cierra el ciclo. La rutina continúa hasta que se a alcanzado la dirección del último de los datos que acomodar, este último dato se encuentra en los registros 013H y 012H.

5.4.4. Eliminación o edición de datos erróneos

Es conveniente poder corregir errores en la toma de datos, por este motivo se diseñó un método de borrar o de volver a capturar las medidas mal tomadas. La primer rutina que se ejecuta es la de Edición de dato. Esta se encarga de 3 acciones: Iniciar variables, determinar el dato que se quiere modificar y ejecutar la rutina correspondiente.

Primeramente se les asigna un valor inicial a las variables que son necesarias tanto para la toma de medidas como para el borrado de dato. En el registro 01AH se almacena el número de archivo del dato a modificar; en los registros 018H y 019H el número de dato en el archivo y en los registros 01BH y 01CH el número de dato en la memoria menos uno. A estos se les asigna el valor de 1. Luego, es desplegado un letrero, se calculan las direcciones, se lee el valor del dato en la memoria y también se despliega. Se muestra además el

número de archivo y número de dato en números hexadecimales. Entonces se espera a que se accione alguna de las teclas.

Con el método de "menú de opciones" cada tecla ejecuta una distinta función: la tecla 1 incrementa el número de archivo, la 2 incrementa el número de dato y la 3 realiza la edición. Cada una de estas funciones es realizada por las subrutinas: a) incrementos de archivo, b) incremento de número de dato, c) edición y d) medición.

5.4.4.1. Incremento de archivo.

Cuando esta rutina es ejecutada se inician las variables: cuenta de datos totales y cuenta de datos a desplegar. Se lee el número de archivos en memoria y se compara con el dato alcanzado hasta ese momento. De ser igual, es iniciado el contador de archivos y se continúa la operación. Después, se incrementa el contador de archivos y se determina la cantidad de datos que hay hasta el archivo anterior. A continuación se lee y almacena en los registros 01DH y 01EH la cantidad de datos que pertenecen al archivo. Tanto a este contador como a el contador del número de dato en memoria se les resta una unidad para compatibilidad con otras rutinas, como la de toma de medidas, ya que estas incluyen una operación en que incrementa estos datos.

5.4.4.2. Incremento de número de dato.

La rutina toma el número de datos de los registro 01DH y 01EH, lo incrementa en una unidad y lo compara con el contenido de los registro 018H y 019H. De ser iguales, carga a estos últimos el valor 0001H para reiniciar el conteo. También se ejecuta el conteo de número de datos hasta el archivo anterior. Cuando no son iguales se incrementan los pares de registros 018H con 019H y el 01BH con 01CH, para mantener el conteo del número de dato en el archivo y el número de dato en la memoria.

5.4.4.3. Edición.

Cuando ya se ha encontrado el dato a modificar, se procede a la edición. Con el método de menú de opciones se determina si el dato va a ser borrado (tecla 1) o si se desea realizar de nuevo la medición (tecla 2). La tecla 3 realiza la función de "abortar la operación", en el caso de que se decida no llevar a cabo ninguna de las alternativas.

Para realizar el borrado, se determina en la memoria la dirección en la cual sobrescribir, a partir del número de dato que se trate. Después, se captura el número total de archivos y con este se determina la dirección del último dato. Se incrementa dos veces la dirección del dato a borrar, para encontrar el primer dato que será transferido y se ejecuta entonces la rutina de "borrar parte" de los datos (que se explica en el inciso (c) de la sección anterior). Por último se almacena el nuevo encabezado de archivo con un número menor en la cantidad de datos que contiene.

5.4.4.4. Medición.

Esta subrutina requiere: el activar la operación de los contadores, las interrupciones del ratón y de los contadores, desplegar el letrero correspondiente y ejecutar en un ciclo la rutina de despliegue de conteo. Para determinar que ya se ha realizado la medición se verifica si no ha cambiado el contenido en el contador de número del dato en la memoria.

VI. PRESENTACIÓN AL USUARIO

INTRODUCCIÓN.

Generalmente los sistemas requieren de elementos externos que den soporte a las funciones que se realizan. En nuestro caso es importante mencionar la realización del programa de comunicación con la PC y del chasis del equipo puesto que son parte importante del proyecto.

Como ya se ha mencionado, uno de los objetivos de la aplicación del sistema como equipo de medición es la posibilidad de descargar toda la información en una computadora de forma tal que sea posible su estudio estadístico. Además de las capacidades de comunicación que se le había dado al sistema, era necesario el que hubiera un programa encargado de controlar la comunicación con el sistema, y almacenar los datos en un archivo ASCII de forma tal que sea fácil de leer desde cualquier aplicación.

El diseño del chasis busca dos cosas: el proporcionarle al Ictiómetro un adecuado nivel de protección contra el medio ambiente como pueden ser la humedad y la temperatura y el hacerlo manejable y fácil de usar. Se construyó un chasis de acero inoxidable de gran resistencia mecánica y a la corrosión. Se encuentra completamente sellado, evitando así filtraciones de agua que puedan dañar el circuito.

6.1. Diseño del programa comunicación residente en la PC

6.1.1. Características del puerto serial en la PC

El enlace con la PC se realizó a través del puerto serial, el cual cumple con la especificación RS 232C. El puerto de la PC es controlado por un circuito dedicado, que comúnmente es el 8251A. Todos los puertos ocupan una localidad en la memoria, en este caso

el puerto esta direccionado en las lineas de 03F8H a la 03FFH. El puerto debe ser programado para el trabajo que va a realizar, indicándole velocidad, forma de comunicación y tipo de atención. El programa de enlace PC - Ictiómetro debe de controlar este puerto, configurarlo, manejar el flujo de información y almacenar los resultados. Debemos recordar que la comunicación se fijó como asíncrona, con 1 bit de inicio, 8 bits de datos, 1 bit final y sin bit de paridad.

6.1.2. Manejo de interrupciones

La diferencias entre las distintas máquinas que indican ser "compatibles" con IBM, nos obligaron a usar las rutinas del BIOS para realizar la comunicación. De esta forma las diferencias entre las máquinas son corregidas por estas rutinas. Por ejemplo, si se tienen diferencias en la forma de operación, de configuración, de distinta ubicación en el mapa del sistema, o alguna otra razón, la rutina del BIOS adecúa la petición con la características particulares de la máquina. Gracias a este nivel de compatibilidad. El programa desarrollado puede ser usado en un mayor número de máquinas, sin que le afecten las diferencias.

La interrupción de software 014H es la encargada de la realización de los servicios de comunicación. Cuando es generada la interrupción le son pasados los datos de la operación por medio del registro A y el DX. En la parte alta de este registro de 16 bits, se encuentra generalmente el tipo de función a ejecutar, mientras que en la parte baja se encuentran los parámetros con los que trabajará. Esta interrupción cuenta con 4 funciones que son:

- 1) Función 00H: Inicialización del puerto de comunicación.

Esta especifica los parámetros de la comunicación, configurando al puerto:

AH = 00H (Función)

AL =

| | |
|---------------------|----------------------------|
| BIT 7, 6 Y 5 | TASA DE BAUDS |
| 000 | 110 |
| 001 | 150 |
| 010 | 300 |
| 011 | 600 |
| 100 | 1200 |
| 101 | 2400 |
| 110 | 4800 |
| 111 | 9600 |
| BITS 4 Y 3 | PARIDAD |
| x0 | NINGUNA |
| 01 | IMPAR |
| 11 | PAR |
| BIT 2 | BITS DE PARO |
| 0 | 1 |
| 1 | 2 |
| BITS 1 Y 0 | LONGITUD DE PALABRA |
| 10 | 7 BITS |
| 11 | 8 BITS |

DX; Número de puerto serial: 0 para el puerto COM1; 1 para COM2

2) Función 01H: Envío de caracteres al puerto.

AH 01H

AL= Caracter enviado.

DX = Número de puerto (0=COM1, 1=COM2).

3) Función 02H: Recibe caracteres del puerto.

AH 02H

DX = Número de puerto (0=COM1, 1=COM2).

4) Función 03H: Regresa el estado del puerto.

AH 03H

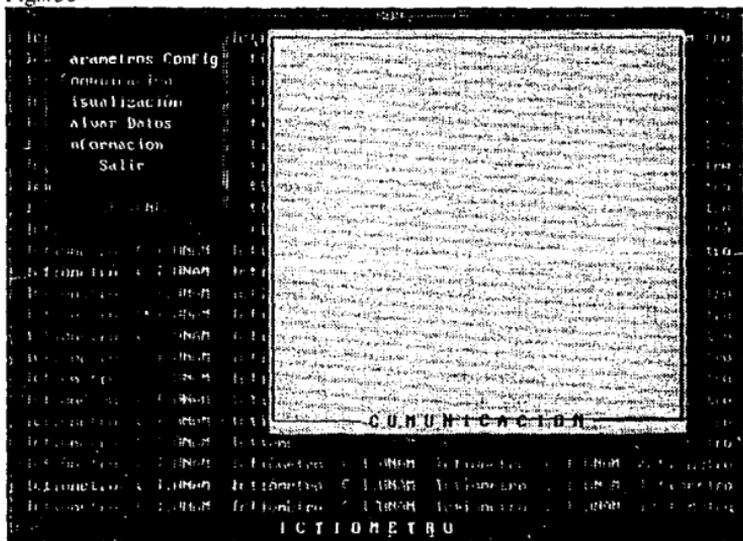
DX = Número de puerto (0=COM1, 1=COM2).

La función regresa:

| AH | INDICACIÓN |
|-------|---|
| BIT 7 | Tiempo Fuera. |
| BIT 6 | Registro de corrimiento de transmisión vacío. |
| BIT 5 | Registro de retención de transmisión vacío. |
| BIT 4 | Detección de rompimiento. |
| BIT 3 | Error de marco. |
| BIT 2 | Error de paridad. |
| BIT 1 | Exceso de ejecución. |
| BIT 0 | Dato listo. |
| AL | INDICACIÓN |
| BIT 7 | Detección de la señal de la línea de recepción. |
| BIT 6 | Indicador de timbre. |
| BIT 5 | Establecimiento del dato listo. |
| BIT 4 | Listo para enviar. |
| BIT 3 | Detección de un cambio en la señal de la línea de recepción. |
| BIT 2 | Detección del borde final del timbre. |
| BIT 1 | Cambio en la señal de estado de la señal de establecimiento del dato. |
| BIT 0 | Cambio en el estado de la señal de listo para enviar. |

El uso de las rutinas del BIOS tiene desventajas: el acceso a los puertos se hace lento; ya que es necesario ejecutar la interrupción y realizar la rutina interna. Esto provoca que sea mucho más lenta la ejecución de la comunicación por la PC que por el lctiómetro. Cuando el lctiómetro envía un dato, tiene que realizar varios ciclos de espera antes de que la computadora pueda ser capaz de recibir el siguiente dato. Este retraso varía de un sistema a otro, dependiendo de la velocidad de la máquina y lo eficiente de la rutina del BIOS.

Fig.#50



6.1.3. Programa de Comunicación en la PC

El problema de la comunicación en la PC se resume a la creación de un programa fácil de manejar para un usuario sin experiencia. El programa tiene que controlar el puerto serial, realizar la comunicación y almacenar los datos.

El programa se diseñó con las siguientes características:

- Programado en lenguaje C.
- En modo de texto.
- Su interface esta hecha con base en ventanas y cajas de diálogo.

El programa cuenta con 4 funciones principales: Configuración, Enlace y Comunicación, Visualización de datos y Salvamento de datos.

Configuración. Para hacer al sistema PC más dinámico, este puede ser configurado de diversas formas de acuerdo a sus propias necesidades. La comunicación se puede realizar por cualquiera de los dos puertos con los que generalmente cuenta un sistema PC. La velocidad de transmisión puede ser fijada en una de las 3 tasas: 1200, 2400 y 9600 bauds.

Enlace y Comunicación. Se ejecuta la comunicación de acuerdo a los parámetros anteriores. De acuerdo al protocolo de comunicación, se envían los códigos de enlace y se realiza la descarga de datos en forma automática. En el caso de que la memoria del lctiómetro indique un posible error, se detendrá la operación y se decidirá si se realiza o no el salvamento de la memoria.

Visualización de datos. Ya que se han descargado los datos, estos pueden ser visualizados (fig. #51) para saber si la transmisión fue bien realizada. Para hacerlo, es necesario dar entrada al número de archivo que se desee verificar. La visualización se hará de acuerdo al tipo de archivo que se trate: largos, anchos, o largos y anchos. Además, se presenta en pantalla la conversión del dato a centímetros.

Salvamento de datos. La última opción es la encargada de la creación de un archivo que contenga todos los datos. La información tiene como encabezado el número de archivo en el lctiómetro, el tipo de archivo que se trata y las medidas en pulgadas y en centímetros. Este archivo esta escrito en código ASCII, por lo que es muy sencillo que se importe en otros programas, como pueden ser hojas de cálculo.

Fig.#51

| dato | LARGO (Pgás) | LARGO (Cms) |
|------|--------------|-------------|
| 1 | 1.01 | 2.5654 |
| 2 | 3.02 | 7.6708 |
| 3 | 5.03 | 12.776 |
| 4 | 7.04 | 17.882 |
| 5 | 9.05 | 23.063 |
| 6 | 1.01 | 2.5654 |
| 7 | 3.02 | 7.6708 |
| 8 | 5.03 | 12.776 |
| 9 | 7.04 | 17.882 |
| 10 | 9.05 | 23.063 |

D A T O S

I C T I O M E T R O

6.1.4. Optimización de la comunicación.

Debido a los retrasos que son necesarios para poder realizar la comunicación entre el lctiómetro y la PC, se creó una rutina que permite determinar cual es el retraso óptimo. La rutina inicia la comunicación con el lctiómetro exactamente de la misma forma en que lo hace para recibir datos, pero envía una señal una vez que se ha establecido el enlace. El código le indica al lctiómetro que en vez de continuar con los datos transmita un patrón de 3 datos, separados por un retraso pequeño. El programa en la PC espera recibir estos datos. Si después de un tiempo límite no los ha recibido, o los recibe mal, el programa envía un código de error, por lo que el lctiómetro incrementa la duración del retraso y repite la operación. Cuando los datos se reciben correctamente, se envía la confirmación. Por último,

el Ictiómetro informa, el número de esperas que se contabilizaron; dato que almacena la computadora para que cada vez que se realice una descarga de datos se le indique al Ictiómetro cuanto debe esperar. La rutina de optimización es ejecutada cada vez que se escoge por primera vez un tasa de transmisión. El dato del número de retrasos necesarios es almacenado en un archivo de configuración, por lo que no es necesario repetir el proceso.

6.2. Diseño y construcción de chasis y de regla de medición.

La construcción física es parte importante del desarrollo, de ella depende la protección, fácil manejo y apariencia. La necesidades de la parte electrónica se reflejan en la forma y funcionamiento del chasis y de la regla de medición; de la misma manera, estos provocaron modificaciones en la operación del Ictiómetro.

6.2.1. Diseño del Chasis.

El chasis es, sobre todo, el elemento de protección del sistema electrónico; de la eficacia con la que cumpla este requerimiento depende la efectividad de todo el aparato. Se definieron las siguientes características con que debía de contar el chasis:

- Soportar la corrosión.
- Evitar el paso de humedad.
- Gran resistencia mecánica.
- Tamaño y peso reducido.
- Facilitar el uso y manejo del Ictiómetro.

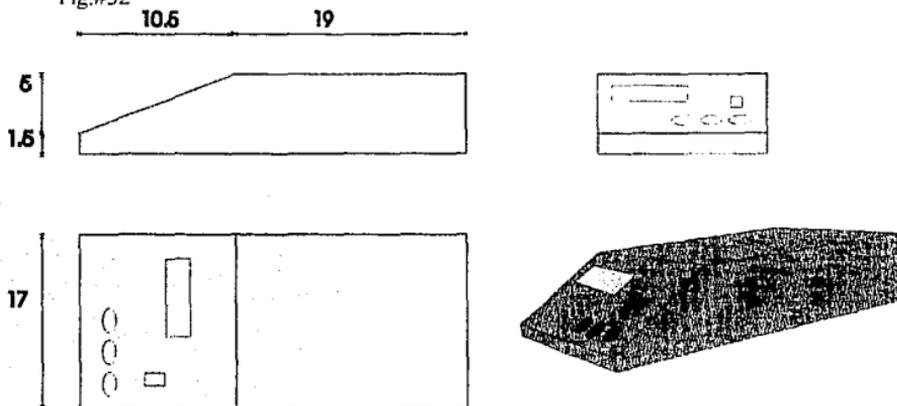
El diseño del chasis fue un proceso evolutivo, primeramente se pensó en construir una mesa que sirviera para realizar la mediciones; en ella estaría el modulo de control que contendría toda la lógica del Ictiómetro. Este módulo podría ser desmontado y conectado a la computadora. Arriba de la mesa estaría el ratón (para realizar la medición), protegido por una pared transparente que evitaría que éste

se mojara fácilmente. Se decidió usar acero inoxidable; debido a la resistencia a la corrosión y mecánica que ofrece; sin embargo, una mesa de acero sería demasiado pesada para ser portátil. Descartada esta opción, el equipo se diseñó en dos partes: un módulo de control y una regla de medición. El primero contendría a la lógica de procesamiento y a la pila (módulo de control); en la segunda estaría el ratón y su guía.

6.2.2. Módulo de control.

El mueble está formado por dos cajas rectangulares montadas una sobre otra. La mas grande guarda los dispositivos lógicos. El diseño se muestra en la figura # 52. El frente esta inclinado para facilitar su lectura y manejo. Ahí se encuentran las 3 teclas de funciones, la pantalla y el interruptor de encendido. En un costado está el puerto del ratón. En la parte posterior el puerto serial, la conexión con un eliminador de batería y el interruptor de reinicio (reset).

Fig.#52



La pila ocupa la segunda caja. En su parte posterior tiene una puerta para substituir la pila fácilmente.

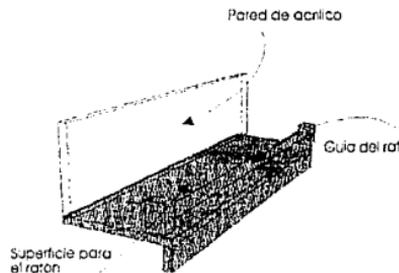
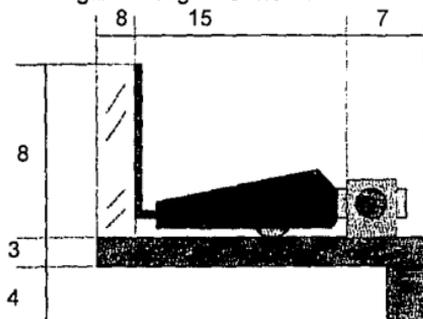
El acceso a los dispositivos se realiza por la parte inferior del chasis, desmontando la base; de esta forma se puede hacer la colocación del circuito, reparaciones, etc. Las tarjetas se colocan una encima de la otra y se sostienen del techo del chasis. Sin contar la base, toda la caja está formada en una sola pieza. La unión entre la base y el resto del chasis cuenta con un empaque de goma para proteger, lo mejor posible, contra la humedad.

6.2.3. Diseño de la regla de medición.

La regla de medición es una superficie rectangular de 15 x 50 cm., en el que se apoya el ratón para moverse. Tiene forma de "L", de modo que se pueden fijar en el costado de una mesa; cuenta con ventosas para adherirse firmemente su superficie. Para garantizar una buena fricción entre el plano y el ratón, se le agrego una superficie rugosa.

El ratón se mueve en una sola dirección gracias a una guía (fig. #53), de esta forma se evita que el dispositivo gire sobre su eje, lo cual provocaría errores en la medición. La guía está formada por una

Fig.#53 Regla de medición



barra circular de acero inoxidable, en la que se monta un buje unido al ratón. También contiene una pantalla de acrílico que le sirve de protección al ratón y a la regla de medición contra suciedades salpicaduras.

VII. CONCLUSIONES.

7.1. Conclusiones del Proyecto.

El proyecto tuvo 2 objetivos principales que fueron alcanzados:

1. La construcción de un equipo capaz de medir y almacenar las dimensiones de un animal marino y
2. La creación de un sistema mínimo de procesamiento digital que permita el desarrollar proyectos de distinta índole.

En cuanto a su desempeño en la aplicación como medidor de peces y de acuerdo a los resultados obtenidos en las pruebas al prototipo se cumplieron y mejoraron las siguientes características planteadas por el Instituto de Ciencias del Mar y Limnología:

- Se alcanzó una resolución teórica de 1/200 de pulgada, (0.1 mm. aprox.), aumentando el rango de operación de 0 a 50 cm.
- La capacidad de almacenamiento es de hasta 4000 datos, los cuales son guardados en forma no volátil, sin la necesidad de mantener alimentado el sistema.
- Gracias a la utilización de la tecnología CMOS y a la posibilidad de poner a la arquitectura en estado de hibernación (antes de realizar la selección de una opción del menú por parte del usuario) podemos reducir el consumo de energía obteniendo una duración en uso continuo de hasta aproximadamente 3 semanas.
- La computadora cuenta con un programa de comunicación amigable, que hace fácil la transferencia de la información. El programa almacena la información en formato ASCII.
- El aparato en forma general es fácil de utilizar: la pantalla facilita su manejo, desplegando las distintas opciones y evitando confusiones.

El sistema de procesamiento digital desarrollado en base al microcontrolador 80C31 tiene las capacidades de:

- Fácil conexión con otros dispositivos.
- Posibilidad de reprogramación.
- Despliegue alfanumérico.
- Interface de comunicación serie con una computadora
- Bajo consumo de energía.

Además, el sistema cuenta con las siguientes características:

a. Se logró que el sistema sea pequeño, gracias al uso del 80C31. Debido a su bajo consumo de energía (120 mA) es adecuado para sistemas portátiles.

b. Se consiguió que el sistema sea capaz de enlazarse con otros dispositivos a través de 9 señales de entrada o salida; 4 de estas líneas pueden ser usadas como señales con otros propósitos: 2 como señales de interrupción y 2 como señales para contadores. De esta forma se consigue que el sistema pueda crecer al serle añadidos otros dispositivos.

c. Para permitir el almacenamiento no-volátil, el borrado y la reescritura de la información, se implementó una memoria externa con capacidad de 64 Kbits. La memoria de datos esta implementada con memorias del tipo EEPROM.

d. Con el fin de que el sistema pueda ser enlazado a otros equipos, se le proveyó de una interface de comunicación serial. La interface esta construida a partir del dispositivo ICL 232. Este dispositivo contiene tanto convertidores para la recepción como para la transmisión y requiere de una sola fuente de alimentación a 5 V.

e. Con el fin de aumentar la seguridad en su uso, el sistema cuenta con sistemas de protección ante las posibles fallas en su operación y con la capacidad de reiniciar las operación sin perdida de información.

Todas estas características le permiten al sistema servir para el desarrollo de distintos equipos, principalmente a través de un esfuerzo de programación y con el mínimo de cambios físicos, de acuerdo a la filosofía de Sistemas de Desarrollo por Software.

7.2. Recomendaciones

Es importante además indicar que aunque se cumplieron los objetivos, el sistema puede ser mejorado en distintas áreas:

1. **Programación.** Es posible hacer una optimización de las rutinas, haciéndolas más eficientes en la utilización de los recursos y aumentado su velocidad de procesamiento. Todo esto redundaría en un sistema más rápido y que consume menos energía.

2. **Configuración física de la tarjeta.** Consideramos necesario revisar la distribución de componentes para mejorar la organización del circuito impreso. Otra posible vía para su optimización es la elección de otras tecnologías de mayor integración y de mejores características mecánicas.

3. **Programación remota.** Si se habilita la escritura en la memoria de programa, es posible construir un sistema que, enlazado con una computadora personal, sea fácilmente reprogramado. De esta forma, el sistema sería más versátil, fácil de adaptar y permitiría una más fácil depuración de los sistemas a desarrollar.

4. **Capacidad de crecimiento.** Se debe de considerar el darle a la tarjeta la posibilidad de fácil acceso a las señales de control del microprocesador con el fin de poder interactuar con otras arquitecturas.

5. **Manejo de los errores operativos.** El alineamiento visual entre el objeto y el dispositivo apuntador para tomar la medida es un error en que se puede incurrir frecuentemente. El cansancio y el tedio

pueden empeorarlo, por lo que es importante que se considere alguna modificación que permita garantizar su correcto alineamiento.

6. Resistencia del Ictiómetro al medio ambiente. Aunque el sistema cumple los requerimientos solicitados dentro del entorno de trabajo del investigador, es posible que bajo condiciones extremas a las estipuladas en el diseño, el aparato llegue a ser dañado, especialmente debido a su exposición directa al agua y/o a una mala limpieza de la zona de trabajo. El proteger al Ictiómetro para este grado de riesgo implicaría un aumento elevado de costos, al tener que basarse en otro tipo de soluciones. El sistema desarrollado puede ser mejorado con ligeras modificaciones, tales como la impermeabilización con resinas epóxicas de los circuitos impresos (lo cual reduciría las posibilidades de mantenimiento) y el diseño de un escudo protector para regla de medición.

Con respecto al ratón, que no está diseñado herméticamente puesto que muchas de sus partes deben estar en contacto con el medio. En caso de una mala limpieza de la regla de medición y del ratón, la bola del dispositivo puede llegar a resbalar sin rodar, lo que ocasionaría que se dejara de sensar el movimiento. Otra posible falla es la acumulación de basura en el lector óptico.

VIII. BIBLIOGRAFÍA.

1. INTEL'S 8 AND 16 BITS MICROCONTROLERS
Editado por Intel Corporation, (1990).
2. TECHNIQUES FOR DIGITIZING ROTARY AND LINEAR MOTION.
Precision Measurement Division.
Editado por Dynamics Research Corporation, 1ª Impresión (1976).
3. SENSORS AND TRANSDUCERS
Keith Brindley
Editado por Heinemann Professional Publishing. 1ª Impresión (1988).
4. MICROPROCESSORS AND INTERFACING:
Programming and Hardware
Douglas V. Hall
Editado por McGraw-Hill International Editions. 3ª Impresión (1988)
5. MICROPROCESSORS/MICROCOMPUTERS:
Architecture, Software and Systems
Adi J. Khambata
Editado por John Wiley & Sons. 2ª Edición (1987)
6. MS DOS: Advanced Programming
Michael J. Young
Editado por Sybex. 1ª Edición (1988)
7. MÉTODOS EXPERIMENTALES PARA INGENIEROS
Jack P. Holman
Editado por McGraw-Hill. 4ª Edición (2ª en español) (1986)

8. **SENSORES Y ANALIZADORES**
Harry N. Norton
Editado por Editorial Gustavo Gili, S. A. 1ª Edición (1984)
9. **HANDBOOK OF BATTERIES AN FUEL CELLS**
David Linden
Editado por McGraw-Hill. (1984)
10. **MEASUREMENT SYSTEMS: APPLICATION AND DESIGN**
Ernest O Doebelin
Editado por McGraw-Hill. (1966)
11. **INTERSIL COMPONENT DATA CATALOG**
General Electric-RCA
Editado por General Electric. (1987)
12. **AND LED AND LCD PRODUCTS**
Editado por William J. Purdy Company. (1988)
13. **DIGITAL SIGNAL PROCESSING WITH THE TMS320C25.**
Chassaing R. y Horning D. W.
Editado por John Wiley & Sons. (1990)

REVISTAS

14. **RADIO ELECTRONICS**
What's New in Batteries
Robert Grossblatt
Publicado por Gernsback Publication Inc. Septiembre (1984)
15. **RADIO ELECTRONICS**
Rechargeable Batteries
Robert Grossblatt
Publicado por Gernsback Publication Inc. Octubre (1984)

-
16. **BYTE**
Under the Hood: The Mouse that roared
Roger Alford
Publicado por Mc Graw-Hill. Noviembre (1990)
 17. **BYTE**
The Birth of the Microprocessor
Federico Faggin
Publicado por Mc Graw-Hill. Marzo 1992
 18. **BYTE**
State of the art: Tomorrow's Chips
Bob Ryan, R. M. Burger, W.C. Holton, Trevor Marshall
Publicado por Mc Graw-Hill. Febrero de 1992
 19. **PC MAGAZINE**
Tutor: Bus mice, serial mice, PS/2 mice: How are they alike?
Jeff Prosis
Publicado por Ziff Communication Comp. Mayo 28 1991
 20. **JOURNAL ELECTRONIC ENGINEERING**
Spatial Ultrasonic Sensors Gain Position in FA Robotics
Shiro Makino y Jiro Inoue
Publicado por Depa Publications. Agosto 1989

MANUALES

21. **MEMORY Databook**
Editado por Exel Microelectronics. Inc. 1988
22. **PROGRAMMABLE LOGIC DEVICES: Databook and Design Guide**
Editado por National Semiconductor. 1989
23. **MEMORY: Databook**
Editado por National Semiconductor. 1988

-
- 24 PROGRAMMABLE LOGIC DEVICES: Databook and Design Guide
Editado por National Semiconductor. 1989
- 25 DIGITAL INTEGRATED ELECTRONICS
Herbert Taub y Donald Schilling
Editado por McGraw-Hill. 12ª Edición 1986

IX. APÉNDICE

A. Descripción funcional de los dispositivos apuntadores.

Los dispositivos apuntadores son instrumentos de entrada de comandos a una computadora e indican la realización de una acción: un movimiento, el inicio de una rutina, la posición de un objeto, etc. En general, todos ellos se caracterizan por ser transductores de movimiento, aunque usan distintas técnicas para hacerlo. Entre los diversos tipos de dispositivos apuntadores que existen se encuentran los "ratones", "plumas de luz", tabletas digitalizadoras y pantallas tipo "touch screen".

Los dispositivos apuntadores tipo "ratón", funcionan basados en técnicas distintas: ópticas, mecánicas u optomecánicas [16].

Ratón de funcionamiento óptico. Este tipo utiliza una superficie graduada con una cuadrícula de líneas negras y azules. El ratón tiene dos diodos emisores de luz (LED's) dirigidos hacia la superficie, uno rojo y otro infrarrojo. Los haces de luz son reflejados en la superficie y

Fig. #55 Ratón óptico

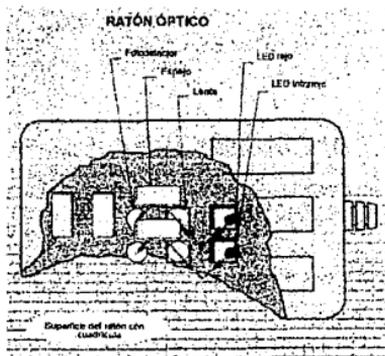
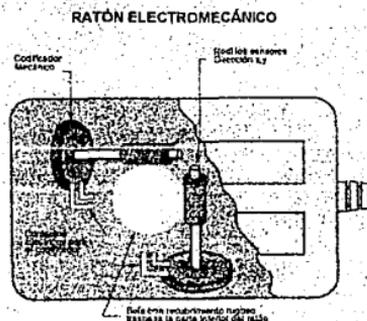


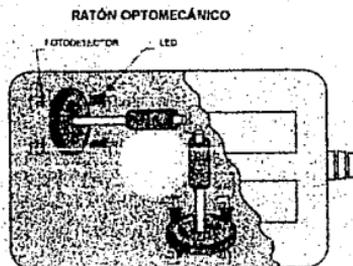
Fig. #54 Ratón electromecánico



regresan al ratón; ahí son enfocados a través de unos lentes para estimular los fotodetectores. Esto se observa en la fig. #55, las líneas azules de la cuadrícula absorben la luz roja, mientras que las líneas negras absorben la infrarroja. De esta forma, mientras el ratón se mueve, la superficie alternativamente absorbe o refleja la luz. Los fotodetectores reconocen la presencia o ausencia de luz, traduciéndolo en señales que envían a la computadora.

Ratón de funcionamiento mecánico. Trabaja por medio de codificadores angulares de movimiento. En este tipo de ratones una bola metálica, con un recubrimiento plástico, atraviesa la parte inferior del cuerpo del dispositivo haciendo contacto con la superficie (como lo muestra la fig. #54). Cuando el ratón se mueve, la bola gira y esta mueve dos rodillos sujetos a los codificadores, que se encargan de digitalizar el movimiento.

Fig. #56 Ratón óptomecánico



Como ya se ha mencionado, existen diversas técnicas para realizar la codificación: electromecánica, optomecánica o magnética; aunque los ratones solo utilizan la electromecánica u optomecánica. En el caso de los ratones electromecánicos, el codificador está compuesto de contactos metálicos que tocan pistas conductoras, con

lo que se transmiten pulsos cuando llegan a encontrarse. Los ratones de tipo optomecánico (fig. #56) utilizan LED's para realizar la codificación. En este caso, un disco con perforaciones en su periferia corta el haz de luz que transmite el diodo; la señal es recibida por un fototransistor, el cual transforma la señal en impulsos eléctricos que posteriormente serán procesados.

Los dispositivos apuntadores tipo ratón también pueden ser clasificados por la clase de interface que requieren para su manejo. Existen actualmente 2 importantes interfaces en los ratones: de puerto serial y de puerto asignado[19].

Ratones de puerto serial. Se caracterizan por que se conectan al puerto serial de una computadora, con la ventaja de que, para su uso, tan solo requieren que se ejecute el software conocido como "manejador del ratón" (driver). Para su comunicación acomodan la información en palabras de control (paquetes) que transmiten a la computadora. Esta información indica cuanto se ha movido el dispositivo desde la ultima transmisión. En el caso del ratón de la compañía Microsoft (MS MOUSE), la información es empaçada en 3 palabras o bytes y las funciones de cada bit se describen en la figura#57.

Fig. #57 Paquetes de datos por bytes en la comunicación serial del MS MOUSE

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|------------|----|----|----|----|----|----|----|----|
| 1º BYTE | X | 1 | LB | RB | Y7 | Y6 | X7 | X6 |
| 2º BYTE | X | 0 | X5 | X4 | X3 | X2 | X1 | X0 |
| 3º BYTE | X | 0 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |

Donde:

LB = Botón Izquierdo: 0 = oprimido o 1 = no oprimido.

RB = Botón Derecho: 0 = oprimido o 1 = no oprimido.

X7-X0 = Indicador del movimiento del ratón de 8 Bits. Cuenta de pasos en dirección horizontal desde la última transmisión.

Y7-Y0 = Indicador del movimiento del ratón de 8 Bits. Cuenta de pasos en dirección vertical desde la última transmisión.

X = Su condición no importa.

La transmisión se realiza a 1200 bps. usando 1 bit de paro y 7 bits de datos (sin paridad).

Su velocidad de transmisión de datos les permite recorrer una distancia de hasta 127 unidades (.67 pulgadas) entre cada transmisión (cerca de 20 milisegundos), sin llegar a perder datos. Estas convenciones varían dependiendo del fabricante; aunque en general, todos mantienen características semejantes.

Ratones de tipo puerto asignado. Se caracterizan porque entregan directamente las señales de los transductores y se conectan a la computadora por medio de una tarjeta de expansión. Las señales son tan sólo filtradas dentro del ratón; por lo que no cuentan con un procesador interno, como en el caso de los seriales. La tarjeta de expansión es la que se encarga de hacer el manejo de señales para determinar la cantidad absoluta de movimiento recorrido por el ratón. En este tipo de arreglos la información es almacenada en registros que son accesados por la computadora.

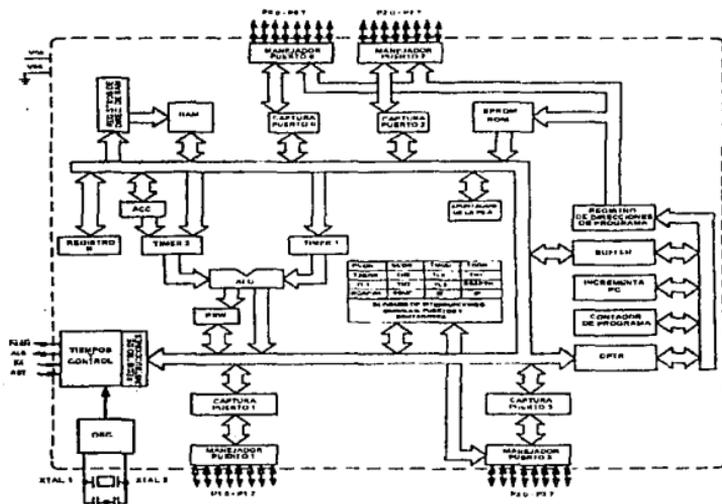
Sin importar el tipo de enlace, el proceso de actualización de la información se lleva a cabo de 30 a 60 veces por segundo. Se realiza a esta velocidad debido a que esta es la velocidad en que la información de la pantalla es refrescada. Aunque este proceso se hiciera un mayor número de veces, la información no sería usada y se perdería en la mayor parte de las ocasiones.

Otro método de intercambio de información es por medio de interrupciones, las cuales son generadas sólo cuando ha habido un cambio en el estado del ratón. Esta forma es muy usada, ya que el CPU no pierde tiempo haciendo un poleo cuando el ratón está detenido.

B. Descripción de la familia de microcontroladores MCS'51.

Los microcontroladores de la familia MCS'51 se caracterizan por tener cada uno 2 contadores de 16 bits, RAM interna, un puerto serial y manejo de la fuente de poder. La arquitectura interna de los dispositivos de la familia de microcontroladores MCS'51 se muestra en al fig. #58 [1].

Fig.#58 Organización interna de la familia MCS'51



1. Registros Internos.

Para su funcionamiento el 80C31 cuenta con veintiún registros de función especial los cuales son:

Acumulador [ACC]. Registro sobre el que operan las instrucciones lógicas y matemáticas.

Registro Auxiliar [B]. Es un registro de apoyo al acumulador, para realizar las operaciones de multiplicación y división. Puede ser usado como un registro común para el resto de las funciones.

Palabra de estado de programa [PSW]. Contiene una serie de banderas (FLAGS) que permiten determinar y manejar el estado de la operación y las condiciones del programa.

Apuntador de la pila [SP]. Es un registro de 8 bits, que indica la dirección de la pila. Este se incrementa cada vez que es usada una instrucción PUSH o CALL. La pila reside en la memoria interna, teniendo un valor inicial de 07H.

Apuntador de datos [DPTR]. Este apuntador, de 16 bits, está formado por dos registros: uno contiene la parte alta (DPH) y el otro la parte baja (DPL), indicando la dirección de datos en la memoria externa. Este registro puede ser manipulado como un solo registro de 16 bits, o como 2 de 8 bits.

Puertos paralelos [Px] 0,1,2,3. Son los registros que representan los accesos a los puertos 0,1,2,3 respectivamente.

Buffer de datos del puerto Serial [SBUF]. Esta formado realmente por dos registros, el buffer de transmisión y el de recepción. Cuando los datos se escriben en él, estos colocados en el buffer de transmisión, donde se mantienen hasta que se realiza la transmisión (de hecho, esto es lo que la inicia). Cuando se realiza una lectura, el dato proviene del buffer de recepción.

Registros de contadores [THx, TLx] 0, 1. Estos pares de registros (TH0, TL0 y TH1, TL1), son los registros usados por los contadores de 16 bits.

Registros de control [IP, IE, TMOD, TCON, T2CON, SCON, PCON]. Estos registros contienen el control y las banderas de estado del sistema de interrupciones, de los contadores/relojes y del puerto serial.

Los registros están organizados en el mapa de memoria interno descrito en la figura#59.

Fig. #59 Mapa de registros

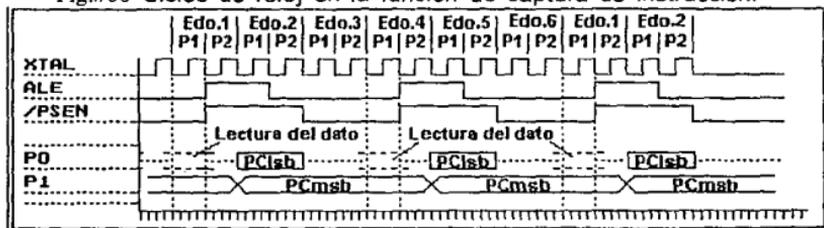
| | | | | | | | | |
|----|------|------|-----|-----|-----|-----|------|----|
| F8 | | | | | | | | FF |
| F0 | B | | | | | | | F7 |
| E8 | | | | | | | | EF |
| E0 | ACC | | | | | | | E7 |
| D8 | | | | | | | | DF |
| D0 | PSW | | | | | | | D7 |
| C8 | | | | | | | | CF |
| C0 | | | | | | | | C7 |
| B8 | IP | | | | | | | BF |
| B0 | P3 | | | | | | | B7 |
| A8 | IE | | | | | | | AF |
| A0 | P2 | | | | | | | A7 |
| 98 | SCON | SBUF | | | | | | 9F |
| 90 | P1 | | | | | | | 97 |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | | 8F |
| 80 | P0 | SP | DPL | DPH | | | PCON | 87 |

2. Descripción Operativa del 80C31.

En esta sección describe la operación de distintas partes del microcontrolador: a) la estructura y operación de los puertos; b) las configuración de entrada/salida de puertos; c) el acceso a la memoria externa; d) los contadores/relojes; e) la interface serial; f) las interrupciones; g) la señal de reinicio; h) el reinicio en el encendido; i) la operación de los modos de ahorro de energía.

Para el manejo de los tiempos de operación del circuito, se han seccionado los períodos de tiempo en los que se realizan las operaciones, de esta forma cada trabajo se realiza en un momento preciso. Estos períodos de tiempo se llaman ciclos de máquina. En el caso del 80C31, un ciclo de máquina ocupa 12 ciclos de oscilación del reloj. Entonces, los procesos que suceden al ejecutarse una instrucción se identifican con el momento determinado en que ocurren dentro del ciclo de máquina. Por lo que, para poder identificar un momento dentro del ciclo de máquina, estos se han dividido en 6 secciones que ocupan 2 períodos de oscilación cada uno. Para indicar este momento se señala su sección (Sx) y su período (Py) dentro del ciclo de máquina. Por ejemplo, la captura de una instrucción tiene el diagrama de tiempos mostrado en la figura #60

Fig.#60 Ciclos de reloj en la función de captura de instrucción.



a) Estructura y operación de los puertos.

La familia de dispositivos MCS'51 cuenta con 4 puertos de tipo bidireccional. Estos 4 puertos comparten otras funciones, además de las de servir de puerto de comunicación directo de datos. Los puertos P0 y P2 son usados como bus de direcciones para el acceso a la memoria externa; además, el puerto P0 sirve como bus de datos, enviando y recibiendo información. Esta función limita el uso de estos puertos como entrada/salida genérica de datos. En el puerto P3, todas las terminales del puerto tienen asignadas otras funciones, que se describen a continuación:

TERMINAL DEL PUERTO

P3.0

P3.1

P3.2

P3.3

P3.4

P3.5

P3.6

P3.7

SEGUNDA FUNCIÓN

RECEPCIÓN DE DATOS DEL PTO. SERIAL (RXD)

TRANSMISIÓN DE DATOS DEL PTO. SERIAL (TXD)

INTERRUPCIÓN EXTERNA 0 (INT0)

INTERRUPCIÓN EXTERNA 1 (INT1)

ENTRADA EXTERNA DEL CONTADOR (T0)

ENTRADA EXTERNA DEL CONTADOR (T1)

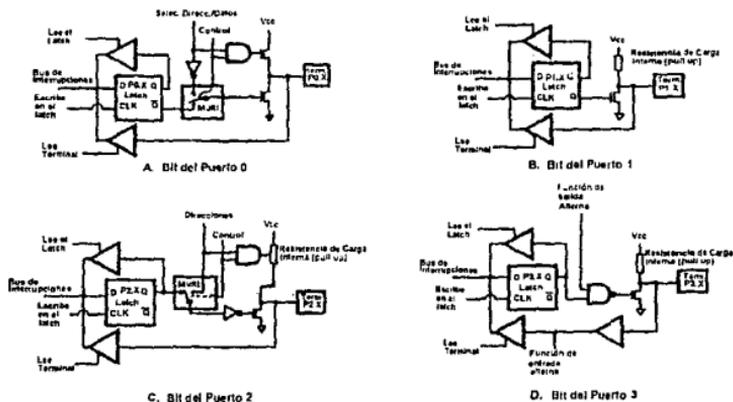
SEÑAL DE HABILITACIÓN DE ESCRITURA (WR)

SEÑAL DE HABILITACIÓN DE LECTURA (RD)

b) Configuraciones de entradas/salidas de puertos.

Cada uno de los puertos tiene diferentes configuraciones de salida (como se muestra en la fig. #61); pero en general todas están formadas por un circuito de captura (latch), un manejador de salida y un circuito de regulación de entrada (buffer). El arreglo de estos elementos varía de un puerto a otro, de acuerdo a su función.

Fig.#61 Operación de los puertos de entrada/salida del microcontrolador



Como se puede observar, los LATCH's conectan tanto sus entradas como salidas al bus interno, por lo que su contenido puede ser leído en cualquier momento. Además, a través de un buffer es posible hacer lecturas directas del contenido de una de las terminales.

Los puertos P1, P2 y P3 tienen salidas con impedancias de carga internas (Pull-up) que manejan la corriente que es drenada del circuito. En el caso del puerto P0, el dispositivo de carga sólo funciona para realizar el acceso a memoria externa. Cuando funciona como puerto, el estado de terminal esta indefinido, o "flota"; y depende de la entrada para tomar un valor. En el caso del puerto P2, cuando se realiza un acceso a memoria externa, la impedancia de carga se ve incrementada por la presencia de una impedancia extra; lo que permite transiciones mucho más rápidas.

Para que los puertos P1, P2 y P3 puedan ser usados como entrada debe ser escrito un 1 en su LATCH; de esta forma, los transistores inferiores del arreglo se cortan y permiten que la señal que entra en la terminal llegue directamente al buffer de lectura. Como se observa en la fig. #61, es posible que estos puertos sean manejados directamente por dispositivos con salida de colector abierto; ya que las impedancias de carga siguen funcionando, en este caso manejando la corriente de entrada.

Algunas instrucciones leen directamente el estado del puerto; otras, por el contrario, leen la salida del LATCH. Las instrucciones que leen el LATCH son las llamadas "leer-modificar-escribir", éstas instrucciones se caracterizan porque leen el contenido del LATCH, modifican el dato y entonces lo escriben de nuevo. La razón de esto es evitar posibles errores. Si se realizara la lectura de la terminal cuando tiene un "1"; por condiciones de carga el voltaje pudiera ser tan bajo que se interprete como otro valor. Si la lectura se hace directamente al LATCH se obtendrá el valor correcto.

c) Acceso a la memoria externa.

Para realizar el acceso de la memoria externa se hace uso de las siguientes señales: \overline{PSEN} (habilitación de almacenamiento de programa) para el acceso a la memoria de programa, \overline{RD} (leer dato) y \overline{WR} (escribir dato) para la memoria de datos. Es necesario el uso de una señal complementaria, conocida como ALE (habilitador de captura de dirección), que se encarga de dirigir el flujo de información; ya que como se mencionó anteriormente, el bus de datos y la parte baja del bus de direcciones se encuentran multiplexados.

Las capturas de códigos de programa siempre se realizan con direcciones de 16 bits, mientras que las de datos pueden ser de 8 o 16 bits. Cuando se realizan capturas de 16 bits el contenido del puerto P2 deja de aparecer en sus terminales, pero se mantiene almacenado en su registro interno; cuando termina la operación este reaparecerá. En las capturas de 8 bits, siempre se mantiene el contenido del puerto P2 en sus terminales. En el caso del puerto P0, el contenido de su registro se modifica todo el tiempo; ya que siempre se escribe un 0FFH cuando se realiza una captura. Por esta razón no es recomendable que se use este puerto cuando se usa memoria externa.

Para que el microcontrolador haga uso de la memoria externa debe existir cualquiera de las siguientes condiciones:

1. La señal \overline{EA} esta activa. Esto indica que no se usará la memoria interna y todas las direcciones son de memoria externa.
2. Cuando se ha rebasado la capacidad de la memoria interna. Si el contador de programa (PC) rebasa la capacidad de memoria interna, la dirección 0FFFH en el caso del 8051, se realizarán capturas a memoria externa. Esto no sucede con el 8031, ya que no tiene memoria interna

d) Contadores/relojes

El 80C31 cuenta con 2 temporizadores de 16 bits, que pueden funcionar como reloj o como contador. Cuando funcionan como reloj su valor se incrementa en cada ciclo de máquina. Recordemos que cada ciclo de máquina son 12 ciclos de reloj, por lo que su exactitud será $1/12$ de la frecuencia de oscilación. Cuando funcionan como contador, su valor se incrementa cuando sucede una transición negativa en sus terminales de entrada: T0 o T1. La terminal es revisada durante S5P2 de cada ciclo de máquina; si ha habido una transición negativa desde la revisión anterior se realiza el incremento. El incremento se realiza en el S3P1 del ciclo siguiente. De acuerdo a esto, la velocidad máxima de conteo es $1/24$ de la frecuencia de oscilación.

Cada contador esta formado a su vez por 2 registros de 8 bits: (TL0) parte baja del contador T0; (TH0) para la parte alta de T0; de la misma forma (TL1) y (TH1) para T1; . Estos registros pueden ser usados para manejar cada uno un byte de la cuenta, como contador independiente o para mantener valores de reinicio de cuenta. Por esto los temporizadores deben ser configurados de distintos modos para adaptarse a las necesidades de la aplicación. De esta forma los registros de cada temporizador son manipulados para realizar la cuenta. El registro que controla la configuración de los temporizadores es TCON. Los modos de operación son:

Modo 0: Cualquiera de los 2 temporizadores funciona como contadores de 13 bits, los 8 bits de TLx y los 5 bits menos significativos de THx.. Generan una interrupción cada vez que cambian de todos unos a todos ceros.

Modo 1: Funcionan igual que en el modo anterior pero con 16 bits.

Modo 2: Contador con recarga. En este modo de operación los contadores trabajan con 8 bits, que son manipulados en el registro (TLx). Cuando se sobrepasa el valor máximo, 0FFH, se genera una interrupción; entonces se carga el

valor contenido en el registro (THx). A partir de este valor se reinicia la cuenta. Este valor debe ser programado y su contenido nunca es cambiado por la operación del contador.

Modo 3: Cada contador funciona de forma distinta en este modo: el contador T1 simplemente mantiene su cuenta como si se hubiera deshabilitado. El contador 2 funciona como 2 contadores separados de 8 bits cada uno. El primero, que es TLO, utiliza las líneas del contador 0; el segundo, TH0, funciona siempre como reloj contando los ciclos de máquina y utiliza las líneas del contador 1: TR1, TF1. Es capaz también de generar la interrupción del contador 1.

e) Interface Serial

El puerto serial cuenta con las características siguientes:

- Tipo "Full Duplex", es decir que puede recibir y transmitir datos al mismo tiempo.
- El manejo de la recepción está separado; es decir, que el dato que se ha recibido se almacena mientras el siguiente dato se termina de recibir. Cuando se ha completado la recepción el nuevo dato es entonces almacenado, perdiendo el anterior.
- El funcionamiento de transmisión y recepción se realiza a través de un solo registro, SBUF. Cuando se escribe en él, la información se almacena en el registro de transmisión. Cuando se lee, se obtiene el contenido del registro de recepción.

El puerto serial se configura en distintos modos de operación, de acuerdo a la forma de transmisión que se va a realizar, así como a la velocidad de transmisión. El registro que controla la configuración del puerto serial es SCON. La velocidad de transmisión es función de la recarga de los temporizadores, por lo que depende de los modos de operación de estos. Para su manipulación se describen a continuación: 1) los modos de operación, 2) el registro de control y 3) la determinación de la tasa de bauds.

(1) Descripción de la operación del puerto serial incluyendo la operación de los distintos modos:

- Modo 0: Transmisión asincrónica de 8 bits de datos y la tasa de transmisión fija a 1/12 de la frecuencia de oscilación del reloj.
- Modo 1: La comunicación es de 10 bits: 1 bit de inicio, 8 de datos y 1 de paro. La frecuencia de transmisión es variable
- Modo 2: En este caso la comunicación es de 11 bits: 1 de inicio, 8 de datos, 1 programable y 1 de paro. En este caso el 9º bit de dato puede ser asignado como 0 o 1 (como puede ser la paridad). Este bit es tomado del contenido del bit RB8, del registro de control SCON. La velocidad de transmisión se puede programar; ya sea como 1/32 o como 1/64 de la frecuencia de oscilación.
- Modo 3: Este modo es igual en operación al modo 2, sólo que en este caso la tasa de transmisión es variable.

En cualquiera de los modos, la transmisión se inicia en cuanto se escribe en el registro de SBUF. La recepción se inicia en cuanto se recibe el primer bit de la transmisión (cuando REN = 1); excepto para el modo 0, que inicia cuando RI = 0 y REN = 1.

(2) Registro de Control puerto serial.

El registro de control del puerto serial se conoce como SCON y tiene las siguientes funciones:



| SM0 | SM1 | MOD0 | DESCRIPCIÓN | TASA DE BAUDS |
|-----|-----|------|-------------------------|-----------------------------|
| 0 | 0 | 0 | Registro de corrimiento | $f_{osc}/12$ |
| 0 | 1 | 1 | 8 bits UART | variable |
| 1 | 0 | 2 | 9 bits UART | $f_{osc}/64$ o $f_{osc}/32$ |
| 1 | 1 | 3 | 9 bits UART | variable |

SM2 Habilita la característica de comunicación multiprocesador.

REN Habilita la recepción de datos. Es manejada por el programa.

TB8 Es el 9º bit que es transmitido en los modos 2 y 3. Manejado por el programa.

RB8 Es el bit que es recibido en los modos 2 y 3. En el modo 1, sin SM2= 0, es el bit de paro.

TI Es la bandera de interrupción de transmisión. Esta es encendida por el circuito cuando se termina la transmisión. Debe ser borrada por el programa.

RI Es la bandera de interrupción de recepción. Esta es encendida automáticamente al terminar de recibir el dato. Debe de ser apagada por el programa.

(3) Tasa de bauds.

Como ya se ha visto, la tasa de bauds (número de caracteres transmitidos por unidad de tiempo), depende del modo de comunicación en el que se trabaje. Esta se debe calcular de acuerdo a las siguientes fórmulas:

Modo 0

$$\text{Tasa de bauds} = \frac{\text{Frecuencia de Oscilación}}{12}$$

Modo 2

$$\text{Tasa de bauds} = \frac{2^{\text{SMOD}}}{64} \times (\text{Frecuencia de Oscilación})$$

Donde:

SMOD = Bit 7 del registro PCON.

Modo 1 y 3

En estos modos es usado el temporizador 1 para generar la tasa de Bauds. El temporizador puede ser usado como reloj o como contador, dependiendo si se usará una base de tiempo interna o externa. En la mayoría de las aplicaciones se utiliza un modo de operación con recarga; para éstos la fórmula a utilizar es:

$$\text{Tasa de bauds} = \frac{2^{\text{SMOD}}}{32} \times \left(\frac{\text{Frecuencia de oscilación}}{12 \times (256 - (\text{TH1}))} \right)$$

Donde TH1 es el valor de recarga del contador 1

f) Interrupciones

En el 80C31 los dispositivos que pueden generar una interrupción de la operación del sistema son los contadores y el puerto serial. Se cuenta además con dos señales externas que sirven para atender demandas de elementos externos al MPU. Todas estas interrupciones son de tipo enmascarable, es decir que su operación puede ser deshabilitada por el programa.

Las señales de interrupción externas, $\overline{INT1}$ y $\overline{INT0}$, pueden ser activadas por detección de un nivel o por transición, dependiendo de lo indicado en los bits IT0 e IT1 del registro TCON. Cuando se activan por nivel, la bandera de la interrupción IEx se activa mientras el estado persista. Cuando son activadas por transición el cambio de un estado 1 a 0 genera la interrupción. La bandera de interrupción es desactivada automáticamente, en cuanto es atendida por el programa.

Cuando se generan las interrupciones de los contadores, son activadas las banderas TF0 y TF1. Estas son también borradas automáticamente cuando se realiza la rutina de atención.

En el caso del puerto serial, la interrupción se genera por cualquiera de las banderas RI o TI. En este caso el programa de atención es el encargado de borrar la bandera; ya que es necesario que se determine cuál de las dos fue la responsable de su generación.

Cualquiera de las banderas de interrupción puede ser encendida o apagada por medio del programa, actuando el MPU en la misma forma que si esta hubiera sido por el circuito.

El registros de habilitación de interrupciones, IE, esta organizado de la siguiente forma:

| | | | | | | | |
|-----|---|---|----|-----|-----|-----|-----|
| MSB | | | | LSB | | | |
| EA | - | - | ES | ET1 | EX1 | ET0 | EX0 |

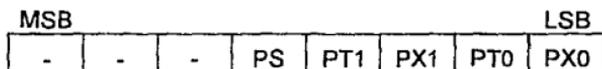
| SÍMBOLO | POSICIÓN | FUNCIÓN |
|---------|----------|---|
| EA | IE.7 | Deshabilita todas las interrupciones cuando es 0; de lo contrario, depende del estado de las habilitaciones individuales. |
| - | IE.6 | Reservado. |
| - | IE.5 | Reservado. |
| ES | IE.4 | Puerto Serial. |
| ET1 | IE.3 | Contador 1. |
| EX1 | IE.2 | Señal de interrupción. externa 1. |
| ET0 | IE.1 | Contador 0. |
| EX0 | IE.0 | Señal de interrupción. externa 0. |

(1) Estructura de nivel de prioridad.

Es posible asignarle a cada interrupción uno de los dos posibles niveles de prioridad. Siempre que se genere una interrupción de prioridad mayor, se dejara el procedimiento en curso para atenderse. Si dos interrupciones de la misma prioridad se generan al mismo tiempo se atenderá primero de acuerdo a una prioridad interna ya asignada que es:

| Nivel | Fuente | Nivel de prioridad |
|-------|---------|--------------------|
| 1 | IE0 | Mayor |
| 2 | TF0 | |
| 3 | IE1 | |
| 4 | TF1 | |
| 5 | RI + TI | Menor |

El registro de prioridad, IP, esta conformado de la siguiente forma:



Si el bit es 1 se asigna una prioridad alta
 Si el bit es 0 se asigna una prioridad baja

| SÍMBOLO | POSICIÓN | FUNCIÓN |
|---------|----------|--|
| - | IP.7 | Reservado |
| - | IP.6 | Reservado |
| - | IP.5 | Reservado |
| PS | IP.4 | Puerto Serial |
| PT1 | IP.3 | Contador 1 |
| PX1 | IP.2 | Señal de interrupción. Externa 1 |
| PT0 | IP.1 | Contador 0 |
| PX0 | IP.0 | Señal de interrupción. Externa 0 |

(2) Manejo de interrupciones.

Las interrupciones son verificadas en S5P2, en todos los ciclos de máquina. De existir alguna, se genera la instrucción de salto a subrutina (LCALL) a la dirección correspondiente al tipo de interrupción. Esta instrucción es bloqueada si:

1. Se genera una interrupción de igual o mayor prioridad.

2. El ciclo de máquina no es el último ciclo que requiere la instrucción para ejecutarse.
3. La instrucción actual es un regreso de interrupción (RETI) o una escritura a los registros IE o IP.

Esto garantiza la correcta operación del programa. Si es detenida la ejecución de la atención a la interrupción, la señal que la generó deberá estar presente durante el siguiente ciclo de máquina para que sea atendida.

Los vectores de interrupción son:

| FUENTE | VECTOR DE DIRECCIÓN |
|---------|---------------------|
| IE0 | 0003H |
| TF0 | 000BH |
| IE1 | 0013H |
| TF1 | 001BH |
| RI + TI | 0023H |

g) Señal de reinicio (Reset).

Para reiniciar la operación del dispositivo, la señal de RST debe de ser activada por lo menos durante 2 ciclos de máquina. Entonces, el MPU genera la señal interna de reinicio a todos los dispositivos. La señal es verificada durante S5P2 en todos los ciclos de máquina. Los puertos mantienen su actividad otros 19 períodos de oscilación después de la señal.

Las señales de ALE y PSEN (Adress Latch Enable o habilitación de la captura de dirección, y Program Store ENable o habilitación del almacenamiento de programa) se mantienen en alto mientras la señal RST esté activa. Después de 1 o 2 ciclos de máquina las señales volverán a funcionar.

Los valores que toman los registros después de la señal de reinicio son:

| Regis tro | Valor |
|--------------|-------------------|
| PC | 0000H |
| ACC | 00H |
| B | 00H |
| PSW | 00H |
| SP | 07H |
| DPTR | 0000H |
| P0-P3 | FFH |
| IP | XXX00000B |
| IE | OXX00000B |
| TMOD | 00H |
| TCON | 00H |
| TH0 | 00H |
| TL0 | 00H |
| TH1 | 00H |
| TL1 | 00H |
| SCON | 00H |
| SBUF | INDETERMINAD O |
| PCON | OXXX0000B |

h) Reinicio en el encendido

Cuando se enciende el circuito es recomendable tener un circuito que genere la señal de inicialización, con el objeto de que el dispositivo no inicie su operación desde una localidad indeterminada de memoria y con sus registros con información errónea. El circuito debe de mantener activada la señal de encendido por lo menos 2 ciclos de máquina.

i) Operación de los modos de ahorro de energía.

Con el fin de ahorrar energía, los dispositivos de tecnología CHMOS cuentan con 2 modos que reducen el consumo: Paro de operaciones y Reducción de Energía. Estos dos modos son controlados a través del registro PCON.

(1) Paro de operaciones.

Cuando se escribe un 1 al bit PCON.0 se entra inmediatamente al modo de Paro de operaciones. En este modo la señal del reloj se desconecta del CPU, pero no del control de las interrupciones, contadores y puerto serial. El CPU mantiene entonces los estados de los registros internos, puertos, RAM, etc. Las señales de ALE y PSEN se mantienen en nivel 1. Los registros GF0 y GF1 funcionan como banderas que pueden ser usadas para determinar las condiciones en las que se generó el estado.

El dispositivo puede salir de este estado por la generación de una interrupción o por la señal de reinicio (RST). Cuando se genera una interrupción, el contenido del bit PCON.0 es borrado automáticamente. Se ejecuta entonces la rutina de atención a la interrupción y cuando se regrese de ella, se continuará la operación en la instrucción que sigue a la que generó el paro de operaciones[1].

Cuando se ejecuta la señal de RST se limpia automáticamente el bit PCON.0; pero se alcanzan a ejecutar algunas instrucciones antes de que el CPU genere la señal interna de reinicio. Este proceso puede causar dificultades al ejecutar algunas instrucciones en forma parcial. Por lo tanto, se recomienda que las instrucciones que siguen a un paro de operación no hagan uso de la memoria externa.

(2) Reducción de energía

Cuando se activa el bit PCON.1 se entra al modo de reducción de energía. Cuando esto sucede, el reloj es desconectado de todo el dispositivo, pero los registros mantienen sus valores, así como la RAM y los puertos. Las señales de ALE y $\overline{\text{PSEN}}$ son fijadas en un nivel bajo.

La única salida de este modo es la generación de la señal RST. Al ejecutarse, se pierde el contenido de los registros internos, pero no de la RAM.

Cuando se encuentra en estado de reducción de energía, el voltaje de alimentación puede ser reducido hasta 2V. Obviamente se debe de garantizar que el voltaje no es reducido sino hasta después de haber entrado en este modo. De la misma forma, la señal de reinicio, que sirve para activarlo, debe de ser generada después de que el voltaje haya regresado a su valor normal y se haya mantenido ahí el tiempo suficiente para normalizar la operación del oscilador.

C. Lenguaje Ensamblador y Programación de la familia MCS'51.

1. Estructura de memoria.

Como se explicó en el apéndice A, el 80C31 tiene un mapa de memoria doble manejando en forma independiente la información de programa y de datos. Esto permite configurar ambos mapas independientemente, además de aumentar la capacidad del dispositivo. Cada mapa de memoria tiene sus propias características y formas de direccionamiento que se explican a continuación.

a) Memoria de Programas.

La primera parte de la memoria de programa, de la 0000H a la 1000H, es direccionada internamente para mantener compatibilidad con los dispositivos de la familia que tienen memoria interna (ROM o EPROM). A partir de la dirección 1000H, todos los dispositivos accesan a la memoria externa de programa. Si se quiere hacer uso de memoria externa desde la dirección 0000H, debe de usarse la señal \overline{EA} para habilitarla.

b) Memoria de datos.

La memoria de datos se encuentra dividida en distintas secciones, de las que depende su forma de direccionamiento. La memoria de datos externa, sólo puede ser direccionada indirectamente, mientras que la memoria interna puede ser direccionada de ambas formas. Internamente se pueden direccionar hasta 256 bytes, debido a que las direcciones son de 8 bits. La primera parte del mapa de datos interno puede ser dividida de la siguiente forma:

Mapa de memoria de datos internos.

| |
|-------------------------------------|
| [7FH] |
| [20H - 2FH] DIRECCIÓN POR BIT |
| [18H - 1FH] BANCO 11 |
| [10H - 17H] BANCO 10 |
| [08H - 0FH] BANCO 01 |
| [00H - 07H] BANCO 00 |

Como se observa, la parte baja de la memoria se encuentra dividida en bancos que pueden ser direccionados en forma directa cuando se encuentran activos. Es posible también accederlos por medio de su dirección absoluta.

De la dirección 20H a la 2FH existe una zona en el que se puede hacer un direccionamiento por bit, permitiendo su uso como banderas o en operaciones booleanas. La última parte es de registros de uso genérico.

Los últimos 128 bytes (de la dirección 80H a la FFH) es un zona de registros dobles, es decir, que sus accesos son distintos dependiendo del tipo de direccionamiento que se realiza: directo o indirecto. Con esto, la capacidad de almacenamiento interno es realmente de 384 bytes. Cuando el direccionamiento es directo se puede acceder los registros de funciones especiales (RFS) y registros de memoria libre. Si el direccionamiento es indirecto se accesa un banco independiente de registros de 128 bytes, que comúnmente son usados como almacenamiento en pila (stack)

| | |
|---|-------------------------------------|
| [FFH] | [FFH] |
| Acceso por direccionamiento indirecto | Acceso por direccionamiento directo |
| [80H] | Registros de funciones especiales |
| | [80H] |
| [7FH] | |
| Acceso por direccionamiento indirecto y directo | |
| [00H] | |

2. Registros de Control

La familia de controladores MCS'51 utiliza una serie de registros para manejar su operación, programando sus funciones, y mostrando el estado de operación del procesador. Estos registros se muestran en la siguiente tabla:

| SÍMBOLO | NOMBRE | DIRECCIÓN |
|---------|---------------------------------|-----------|
| ACC | ACUMULADOR | 0E0H |
| B | REGISTRO B | 0F0H |
| PSW | PALABRA DE ESTADO DEL PROGRAMA | ODOH |
| SP | APUNTADOR DE LA PILA | 81H |
| DPTR | APUNTADOR DE DATOS (2 BYTES) | |
| DPL | PARTE BAJA | 82H |

| | | |
|------|---|------|
| DPH | PARTE ALTA | 83H |
| P0 | PUERTO 0 | 80H |
| P1 | PUERTO 1 | 90 |
| P2 | PUERTO 2 | 0A0H |
| IP | CONTROL DE PRIORIDAD DE INTERRUPCIONES | 0B8H |
| IE | CONTROL DE HABILITACIÓN DE INTERRUPCIONES | 0A8H |
| TMOD | CONTROL DE MODO DEL CONTADOR/RELOJ | 89H |
| TCON | CONTROL DEL CONTADOR/RELOJ | 88H |
| TH0 | PARTE ALTA DEL RELOJ 0 | 8CH |
| TL0 | PARTE BAJA DEL RELOJ 0 | 8AH |
| TH1 | PARTE ALTA DEL RELOJ 1 | 8DH |
| TL1 | PARTE BAJA DEL RELOJ 1 | 8BH |
| SCON | CONTROL SERIAL | 98H |
| SBUF | BUFFER DE DATOS SERIALES | 99H |
| PCON | CONTROL DE ALIMENTACIÓN | 87H |

Aunque algunos de estos registros ya han sido explicados en el texto principal, mencionaremos sus características más importantes.

El registro ACC es el registro sobre el que se realiza las operaciones lógicas y matemáticas; el registro B es usado en la multiplicación y la división, aunque puede ser usado como almacenamiento.

Para el uso de los programas, el registro de estados PSW contiene banderas que indican las condiciones y los resultados de las

operaciones del programa. El registro esta conformado de la siguiente forma:

| | | | | | | | |
|-----|----|----|-----|-----|----|-----|---|
| MSb | | | | | | LSb | |
| CY | AC | F0 | RS1 | RS0 | OV | - | P |

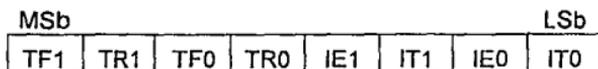
| SÍMBOLO | POSICIÓN | FUNCIÓN |
|---------|----------|----------------------------------|
| CY | PSW.7 | Bandera de acarreo |
| AC | PSW.6 | Auxiliar del Acarreo |
| F0 | PSW.5 | Bandera de propósito general |
| RS1 | PSW.4 | Selección del banco de registros |
| RS0 | PSW.3 | Selección del banco de registros |
| OV | PSW.2 | Bandera de sobreflujo |
| - | PSW.1 | Bandera definible |
| P | PSW.0 | Paridad del Acumulador |

Los registros SP, PC, DPTR, sirven para apuntar posiciones en la memoria sobre las que se realizarán los procesos de almacenamiento en pila, lectura de instrucción y lectura de datos respectivamente.

Para acceder dispositivos internos tales como puertos, y contadores se cuenta con registros como los Pn cuya función es accesos a los puertos de salida; debemos recordar que el acceso a los puertos tiene algunas restricciones que se mencionaron en el texto principal.

Por otra parte, los contadores muestran el estado de la cuenta a través de los registros THn y TLn. Para el manejo de las condiciones de operación de los contadores se utilizan los registros TMOD y TCON. En el caso del registro TCON también es usado para programar la forma de procesar las interrupciones. Estos registros están organizados de la siguiente forma:

TCON



| SÍMBOLO | POSICIÓN | FUNCIÓN |
|---------|----------|---|
| TF1 | TCON.7 | Bandera de sobreflujo del contador 1 |
| TR1 | TCON.6 | Bit de control de actividad del contador 1 |
| TF0 | TCON.5 | Bandera de sobreflujo del contador 0 |
| TR0 | TCON.4 | Bit de control de actividad del contador 0 |
| IE1 | TCON.3 | Bandera de detección de interrupción 1 |
| IT1 | TCON.2 | Control de tipo de disparo de la interrupción 1: Por nivel o transición |
| IE0 | TCON.1 | Bandera de detección de interrupción 0 |
| IT0 | TCON.0 | Control de tipo de disparo de la interrupción 0: Por nivel o transición |

TMOD

| MSb | | | | LSb | | | |
|----------------|-----|----|----|----------------|-----|----|----|
| TEMPORIZADOR 1 | | | | TEMPORIZADOR 0 | | | |
| GATE | C/T | M1 | M0 | GATE | C/T | M1 | M0 |

| SÍMBOLO | FUNCIÓN |
|---------|--|
| GATE | Bandera de sobreflujo del contador 1 |
| C/T | Bit de control de actividad del contador 1 |
| M1 | Bandera de sobreflujo del contador 0 |
| M0 | Bit de control de actividad del contador 0 |

| M1 | M0 | MODO DE OPERACIÓN |
|----|----|---|
| 0 | 0 | 0 Reloj de 13 bits |
| 0 | 1 | 1 Contador o reloj de 16 bits |
| 1 | 0 | 2 Contador o reloj de 8 bits con recarga automática |
| 1 | 1 | 3 Temporizador 0. El registro TL0 funciona como un contador o reloj independiente de 8 bits, y su operación es controlada por los bits de control 0; La parte alta TH0 funciona como un reloj usando los bits de control del temporizador 1 |
| 1 | 1 | 3 Temporizador 1 detiene su operación |

El puerto serial se maneja a través del registro SCON. Además, existe un registro llamado SBUF, que sirve para acceder el dato recibido, o a través de él, transmitir algún dato. Como se mencionó en el capítulo IV para generar el baudaje de la transmisión se utiliza el temporizador 1.

SCON

| MSb | | | | LSb | | | |
|-----|-----|-----|-----|-----|-----|----|----|
| SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |

| SÍMBOLO | POSICIÓN | FUNCIÓN |
|---------|----------|---|
| SM0 | SCON.7 | 1º bit de control de modo de operación del puerto |
| SM1 | SCON.6 | 2º bit de control de modo de operación del puerto |
| SM2 | SCON.5 | Habilita la operación de comunicación multiprocesador en los modos 2 y 3. |
| REN | SCON.4 | Habilitación de la recepción |
| TB8 | SCON.3 | 9º bit transmitido en los modos de operación 2 y 3 |
| RB8 | SCON.2 | 9º bit recibido en el modo 2 y 3 |
| TI | SCON.1 | Bandera de interrupción por fin transmisión. |
| RI | SCON.0 | Bandera de interrupción por fin recepción. |

Por último el registro de control de alimentación (PCON) tiene funciones distintas:

PCON

| MSb | | | | LSb | | | |
|------|---|---|---|-----|-----|----|-----|
| SMOD | - | - | - | GF1 | GF0 | PD | IDL |

| SÍMBOLO | POSICIÓN | FUNCIÓN |
|---------|----------|--|
| SMOD | PCON.7 | Bit de activación de la duplicación del la tasa de bauds |
| - | PCON.6 | no tiene uso. |
| - | PCON.5 | no tiene uso. |
| - | PCON.4 | no tiene uso. |
| GF1 | PCON.3 | Bit de bandera de uso general |
| GF0 | PCON.2 | Bit de bandera de uso general |
| PD | PCON.1 | Bit de habilitación de estado de baja potencia. |
| IDL | PCON.0 | Bit de habilitación de modo de desocupado. |

3. Modos de direccionamiento.

Dependiendo de la forma en que se hace el acceso a los datos, el conjunto de instrucciones del microcontrolador puede dividirse en:

-Direccionamiento directo. En estas instrucciones el operando es especificado por una dirección de 8 bits, que identifica su campo. Sólo la memoria interna puede ser direccionada directamente.

p.e.: MOV <direct1>, <direct2>

donde <, > indican que el valor numérico representa una dirección
direct1 y direct2 son las direcciones de los registros internos

Esta instrucción copia el contenido de la localidad de memoria "direct2" a la localidad "direct1".

-Direccionamiento indirecto. Las instrucciones que realizan direccionamiento indirecto son en las que el operando es indicado por la dirección que contiene un registro. Tanto la RAM interna como la externa pueden ser indirectamente direccionadas. En el caso de la RAM interna, pueden servir como registro de dirección solamente los registros R0 y R1 del banco seleccionado, así como el apuntador de la pila de almacenamiento (stack). En el caso de la memoria externa, el único registro que puede ser usado es el DPTR, que es de 16 bits.

p.e.: MOV <direct1>, @Ri

donde @ indica que se use el contenido de dicho registro como dirección
Ri es el registro donde esta almacenada la dirección de la localidad de memoria.

Esta instrucción copia el contenido de localidad a la que apunta el registro "Ri" en la localidad "direct1".

-Direccionamiento inmediato. En estas instrucciones el valor de una constante sigue al código en las memoria de programa.

p.e.: MOV <directo>, #dato

donde # indica que se trata de un valor numérico constante "dato" es el valor constante predeterminado.

La instrucción escribe el valor constante "dato" en el registro cuya dirección es "directo".

-Instrucciones con registros. Se puede hacer un acceso directo a los registros del banco activo, usando un código de 3 bits en la instrucción; de esta forma el código es más eficiente, ya que elimina la necesidad de usar un byte de

dirección. El banco de registros activos debe de ser indicado en el PSW.

p.e.: MOV <directo>, Rn
donde Rn es el número de registro del banco activo.

Con esta instrucción el contenido del registro "Rn" se copia en la localidad "directo".

- **Instrucciones específicas para registros.** Algunas instrucciones funcionan sólo con ciertos registros, como puede ser el acumulador o el apuntador de datos. Estas instrucciones no necesitan de un byte de dirección, puesto que el código ya contiene esta información.

p.e.: SWAP A
donde A es el número de registro del acumulador.

Con esta instrucción los cuatro bits menos significativos son intercambiados por los cuatro más significativos del registro "A".

- Direccionamiento indexado.** La memoria de programa es la única que puede ser direccionada a través de un índice. Este tipo de direccionamiento es útil para leer tablas de datos. La dirección puede ser formada por la suma del contenido en el DPTR o el PC más el acumulador. El acumulador sirve entonces de índice de la tabla de datos. Es posible hacer otro tipo de direccionamiento indexado, pero este sólo sirve para hacer saltos tipo "en caso de". En esta instrucción, el contador de programa salta a la dirección que indica la suma de los registros del Acumulador y DPTR continuando con la ejecución desde esa parte del programa. De esta forma se puede leer el contenido de la dirección en la memoria de programa, a la que apunta la suma del contenido del registro contador de programa (PC) más el del acumulador (A) y almacena el dato en el acumulador.

p.e.: MOVC A, @A + PC

4. Juego de Instrucciones.

Las distintas instrucciones con las que cuenta el 80C31, dependiendo de la función que realicen, se pueden dividir en [1]:

Instrucciones Aritméticas:

| MNEMONICO | OPERACIÓN | MODO DE DIRECCIONAMIENTO | | | |
|-----------------|---|----------------------------|-----|-----|-----|
| | | DIR | IND | REG | INM |
| ADD A, <BYTE> | $A = A + \langle \text{BYTE} \rangle$ | X | X | X | X |
| ADDC A, <BYTE> | $A = A + \langle \text{BYTE} \rangle + C$ | X | X | X | X |
| SUBBC A, <BYTE> | $A = A + \langle \text{BYTE} \rangle - C$ | X | X | X | X |
| INC A | $A = A + 1$ | SÓLO EL ACUMULADOR | | | |
| INC <BYTE> | $\langle \text{BYTE} \rangle = \langle \text{BYTE} \rangle + 1$ | X | X | X | |
| INC DPTR | $\text{DPTR} = \text{DPTR} + 1$ | SÓLO EL APUNTADOR DE DATOS | | | |
| DEC A | $A = A - 1$ | SÓLO EL ACUMULADOR | | | |
| DEC <BYTE> | $\langle \text{BYTE} \rangle = \langle \text{BYTE} \rangle - 1$ | X | X | X | |
| MUL AB | $B:A = B \times A$ | SÓLO ACC Y B | | | |
| DIV AB | $A = \text{INT}(A/B); B = \text{MOD}(A/B)$ | SÓLO ACC Y B | | | |
| DA A | AJUSTE DECIMAL | SÓLO EL ACUMULADOR | | | |

Donde:

Dir = Direccionamiento directo

Ind = Direccionamiento indirecto

Inm= Direccionamiento inmediato

Reg= Direccionamiento con registros

C = Bit de acarreo

Instrucciones Lógicas:

| MNEMONICO | OPERACION | MODO DE DIRECCIONAMIENTO | | | |
|-------------------|---------------------------------|--------------------------|-----|-----|-----|
| | | DIR | IND | REG | INN |
| ANL A, <BYTE> | A = A .AND. <BYTE> | X | X | X | X |
| ANL <BYTE>, A | <BYTE> = <BYTE> .AND. A | X | | | |
| ANL <BYTE>, #DATO | <BYTE> = <BYTE> .AND. #DATO | X | | | |
| ORL A, <BYTE> | A = A .OR. <BYTE> | X | X | X | X |
| ORL <BYTE>, A | <BYTE> = <BYTE> .OR. A | X | | | |
| ORL <BYTE>, #DATO | <BYTE> = <BYTE> .OR. #DATO | X | | | |
| XRL A, <BYTE> | A = A .XOR. <BYTE> | X | X | X | X |
| XRL <BYTE>, A | <BYTE> = <BYTE> .XOR. A | X | | | |
| XRL <BYTE>, #DATO | <BYTE> = <BYTE> .XOR. #DATO | X | | | |
| CRL A | A = 00H | SÓLO EL ACUMULADOR | | | |
| CPL A | A = .NOT. A | SÓLO EL ACUMULADOR | | | |
| RL A | ROTAR A LA IZQUIERDA A 1 BIT | SÓLO EL ACUMULADOR | | | |
| RLC A | ROTAR IZQUIERDA A A TRAVÉS DE C | SÓLO EL ACUMULADOR | | | |
| RR A | ROTAR A LA DERECHA A 1 BIT | SÓLO EL ACUMULADOR | | | |
| RRC A | ROTAR DERECHA A A TRAVÉS DE C | SÓLO EL ACUMULADOR | | | |
| SWAP A | INTERCAMBIAR LOS NIBBLES DE A | SÓLO EL ACUMULADOR | | | |

Transferencia de datos:

a) Memoria Interna:

| MNEMONICO | OPERACION | MODO DE DIRECCIONAMIENTO | | | |
|--------------------|---|--------------------------|-----|-----|-----|
| | | DIR | IND | REG | INN |
| MOV A, <FNTE> | A = <FNTE> | X | X | X | X |
| MOV <DEST>, A | <DEST> = A | X | X | X | |
| MOV <DEST>, <FNTE> | <DEST> = <FNTE> | X | X | X | X |
| MOV DPTR, #DATO | DPTR= CONSTANTE INMEDIATA DE 16 BITS | | | | X |
| PUSH <FNTE> | INC SP : MOV @SP, <FNTE> | X | | | |
| POP <DEST> | MOV <DEST>, @SP : DEC SP | X | | | |
| XCH A, <BYTE> | INTERCAMBIO DE ACC Y <BYTE> | X | X | X | |
| XCHD A, @Ri | INTERCAMBIO DE LA PARTE BAJA DE ACC Y @Ri | | X | | |

Donde: dest = registro destino; fnte = registro fuente

b) Memoria Externa:

| MNEMONICO | OPERACION | LONG. DE DIR. |
|---------------|---|---------------|
| MOVX A, @Ri | Lectura externa en la dir contenida en Ri | 8 bits |
| MOVX @Ri, A | Escritura externa en la dir contenida en Ri | 8 bits |
| MOVX A, @DPTR | Lectura externa en la dir contenida en DPTR | 16 bits |
| MOVX @DPTR, A | Escritura externa en la dir contenida en DPTR | 16 bits |

c) Tablas en memoria de programa:

| MNEMÓNICO | OPERACIÓN |
|-------------------|---|
| MOVX A, @A + DPTR | Lectura en memoria de programa en la dirección (A + DPTR) |
| MOVX A, @A + PC | Lectura en memoria de programa en la dirección (A + PC) |

Instrucciones Booleanas:

| MNEMÓNICO | OPERACIÓN |
|--------------------------------|-------------------------------------|
| ANL C, bit | $C = C \text{ .AND. bit}$ |
| ANL C, $\overline{\text{bit}}$ | $C = C \text{ .AND. .NOT. bit}$ |
| ORL C, bit | $C = C \text{ .OR. bit}$ |
| ORL C, $\overline{\text{bit}}$ | $C = C \text{ .OR. .NOT. bit}$ |
| MOV C, bit | $C = \text{bit}$ |
| MOV bit, C | $\text{bit} = C$ |
| CLR C | $C = 0$ |
| CLR bit | $\text{bit} = 0$ |
| SETB C | $C = 1$ |
| SETB bit | $\text{bit} = 1$ |
| CPL C | $C = \text{.NOT. C}$ |
| CPL bit | $\text{bit} = \text{.NOT. bit}$ |
| JC rel | Salta si $C = 1$ |
| JNC rel | Salta si $C = 0$ |
| JB bit, rel | Salta si $\text{bit} = 1$ |
| JNB bit, rel | Salta si $\text{bit} = 0$ |
| JBC bit, rel | Salta si $\text{bit} = 1$; CLR bit |

Instrucciones de control de flujo de programa:

| MNEMÓNICO | OPERACIÓN |
|-----------------|--|
| JMP <dir> | Salto a la dirección <dir> |
| JMP <@A + DPTR> | Salto a la dirección que indica A+DPTR |
| CALL <dir> | Salto a la dirección de la subrutina dir |
| RET | Regreso de subrutina |
| RETI | Regreso de interrupción |
| NOP | No realiza operación |

Salto condicionales:

| Mnemónico | Operación | Modo de direccionamiento | | | |
|-------------------------|----------------------------------|--------------------------|-----|-----|-----|
| | | Dir | Ind | Reg | Inn |
| JZ rel | Salta si A = 0 | Sólo el acumulador | | | |
| JNZ rel | Salta si A ≠ 0 | Sólo el acumulador | | | |
| DJNE <byte>, rel | Decrementa y salta si no es cero | X | | X | |
| CJNE A, <byte>, rel | Salta si A ≠ <byte> | X | | | X |
| CJNE <byte>, #dato, rel | Salta si <byte> ≠ #dato | | X | X | |

Para una mayor información en cuanto a la forma de operación de cada instrucción se recomienda el manual de microcontroladores de 8 y 16 bits de Intel [1].

D. Implantación Física

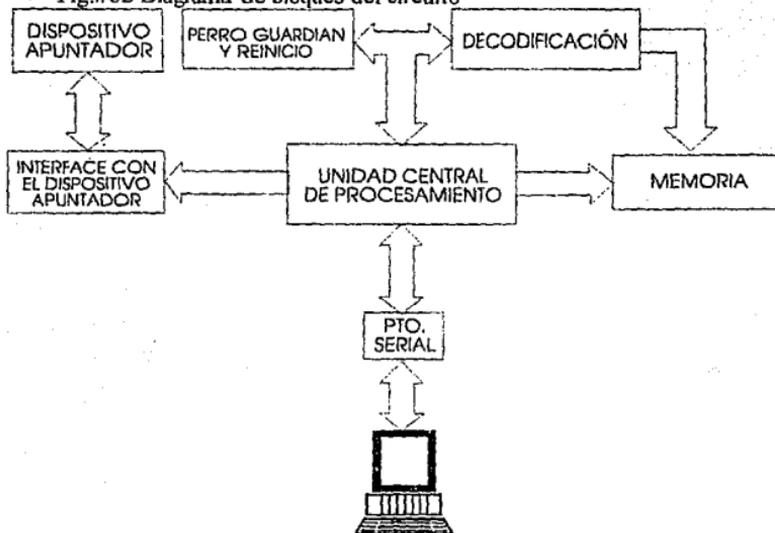
El diseño de los circuitos en diagrama esquemático y de circuito impreso se realizó con la ayuda del paquete ORCAD. El paquete permite obtener un circuito impreso a partir del diagrama esquemático del sistema, por lo que es necesario que todos los elementos del circuito estén completamente definidos en forma, tamaño, nombre y otras características. El diagrama tiene que pasar por una fase de optimización; ya que por limitaciones del programa, este a veces no conecta algunas partes o no lo hace en forma conveniente.

Para realizar el diagrama de circuito impreso fue necesario definir:

3. Cada uno de los elementos y de las conexiones entre ellos.
4. Nombre, dimensiones, forma y conexiones de cada uno de los elementos del circuito:
5. El número de planos en que estaría realizado el circuito.
6. El tamaño de los distintos tipos de pista: señales, alimentación, etc.
7. El tamaño de las conexiones entre los distintos planos del circuito.

Durante la creación del diagrama esquemático (Fig.#62) se definen los distintos nodos que conforman el circuito. A partir de esta definición de conexiones es que se realiza el diseño de las pistas del circuito impreso; por lo que es importante que no hayan errores, como cruces o no-contactos.

Fig.#62 Diagrama de bloques del circuito



Tiene que serle asignado un nombre a cada componente, con el fin de identificarlo. Es importante, sobre todo, que sea fácil reconocer individualmente elementos como resistencia o capacitores; ya que se puede incurrir en errores. Es necesario además asignarles su valor (de tenerlo) o en el caso de que se tengan varios elementos en un mismo empaque, cual de ellos se usará.

Es necesario conocer las características físicas de cada uno de los elementos que componen el circuito y asignarles un tipo específico de empaquetado correspondiente. El programa cuenta con una librería de empaques comunes pero en caso de que el elemento no corresponda a ninguno de ellos es posible definirle uno propio. En el sistema de medición de peces diseñado, la lista de elementos y nombres de dispositivos es la siguiente:

CONEXIÓN BÁSICA MPU 8031

Revised: June 1, 1992

ICTIO-MPU8031-0001 Revision: R3

Bill Of Materials June 1, 1992 12:15:03 Page 1

| Item | Quanti | Reference | Part |
|------|--------|---------------------------------|---------------|
| 1 | 1 | ACO RS1 | ICL232 |
| 2 | 2 | DECOD. MEM.1, ACOND.SENAL1 | PAL20L8 |
| 3 | 1 | BT1 | BATTERY |
| 4 | 1 | CON'TR. VO DSP.1 | TRIM 10 K |
| 5 | 1 | C1 | 10 UF |
| 6 | 2 | C2,C3 | 27 NF |
| 7 | 1 | C4 | .1 UF |
| 8 | 5 | C5,C8,C9,C10,C11 | 22 UF |
| 9 | 2 | C6,C7 | 150 UF |
| 10 | 2 | C12,C13 | 22 PF |
| 11 | 4 | C14,C15,C16,C17 | .0015 NF |
| 12 | 1 | C18 | 25 UF |
| 13 | 1 | C19 | 50 UF |
| 14 | 3 | C20,C21,C22 | 6.9 NF |
| 15 | 1 | D1 | DIODE ZENER1 |
| 16 | 1 | D2 | LED |
| 17 | 1 | JPDIS1 | 300DIP14 |
| 18 | 1 | JP1 | HEADER 2X2 |
| 19 | 1 | JP2 | HEADER 4 |
| 20 | 7 | JP3,JP4,JP5,JP6,JP7,JP8, JP9 | HEADER 6 |
| 21 | 1 | J1 | PHONEJACK |
| 22 | 1 | LATCH1 | 74LS374 |
| 23 | 1 | MCU1 | 80C31 |
| 24 | 1 | MEM DAT1 | 2865 |
| 25 | 1 | MEM PRG1 | 2764 |
| 26 | 2 | MONOEST1,MONOEST2 | 74LS123 |
| 27 | 1 | OR1 | 74LS32 |
| 28 | 1 | PUERTO SERIE1 | CONNECTOR DB9 |
| 29 | 3 | Q1,Q2,Q3 | EM4010 |
| 30 | 1 | REGULADOR1 | LM7805 |
| 31 | 1 | R1 | 6.8 K |
| 32 | 1 | R2 | 1 K |
| 33 | 1 | R3 | 150 OHMS |
| 34 | 4 | R4,R5,R8,R9 | 10 K |
| 35 | 2 | R6,R7 | 100 K |
| 36 | 2 | R10,R15 | 82 K |

| | | | |
|----|---|-----------------|---------------|
| 37 | 4 | R11,R12,R13,R14 | 6.6 K |
| 38 | 3 | R16,R17,R19 | RESISTOR |
| 39 | 1 | R18 | 180 K |
| 40 | 3 | R20,R22,R23 | 1.2 M |
| 41 | 1 | R21 | 2.2 K |
| 42 | 1 | SW1 | SW PUSHBUTTON |
| 43 | 3 | SW2,SW3,SW4 | SW PUSHDB |
| 44 | 1 | WATCH DOG 1 | 74LS423 |
| 45 | 1 | X1 | 12 MHZ |

A estas 45 partes les corresponde un tipo especial de empaquetado, como se muestra a continuación:

CONEXIÓN BÁSICA MPU 8031

Revised: June 1, 1992

ICTIO-MPU8031-0001 Revision: R3

Bill Of Materials June 1, 1992 23:38:45 Page 1

| Item | Quant | Reference | Part |
|------|-------|---|----------|
| 1 | 4 | WATCH DOG 1, MONOEST1, ACOR S1, MONOEST2 | 16DIP300 |
| 2 | 2 | DECOD. MEM.1, ACOND. SENAL1 | 24DIP300 |
| 3 | 2 | SW1, BT1 | TJACK200 |
| 4 | 1 | CONTR. VO DSP.1 | RJ26W |
| 5 | 9 | C1, C2, C3, C4, C5, C8, C9, C10, C11 | CK14 |
| 6 | 2 | C6, C7 | CM05 |
| 7 | 7 | C12, C13, C18, C19, C20, C21, C22 | CK12 |
| 8 | 4 | C14, C15, C16, C17 | CK06 |
| 9 | 2 | D1, D2 | DO7 |
| 10 | 2 | OR1, JPDIS1 | 14DIP300 |
| 11 | 1 | JP1 | 2X2HH100 |
| 12 | 1 | JP2 | 4HH100 |
| 13 | 7 | JP3, JP4, JP5, JP6, JP7, JP8, JP9 | 6HH100 |
| 14 | 1 | J1 | 3HH100 |
| 15 | 1 | LATCH1 | 20DIP300 |
| 16 | 1 | MCU1 | 40DIP600 |
| 17 | 2 | MEM PRG1, MEM DAT1 | 28DIP600 |
| 18 | 1 | PUERO SERIE1 | DB9M |
| 19 | 3 | Q1, Q2, Q3 | TO92 |
| 20 | 1 | REGULADOR1 | TO3B |
| 21 | 23 | R1, R2, R3, R4, R5, R6, R7, R8, | RC12 |

| | | | |
|----|---|---|--------|
| | | R9,R10,R11,R12,R13,R14, R15,R16,R17,R18,R19,R20, R21,R22,R23 | |
| 22 | 3 | SW2,SW3,SW4 | PUSHB |
| 23 | 1 | X1 | HC18UV |

Se definieron 3 tarjetas de circuito impreso a realizar: la tarjeta de sistema de procesamiento, la tarjeta de aplicación y la tarjeta de teclado.

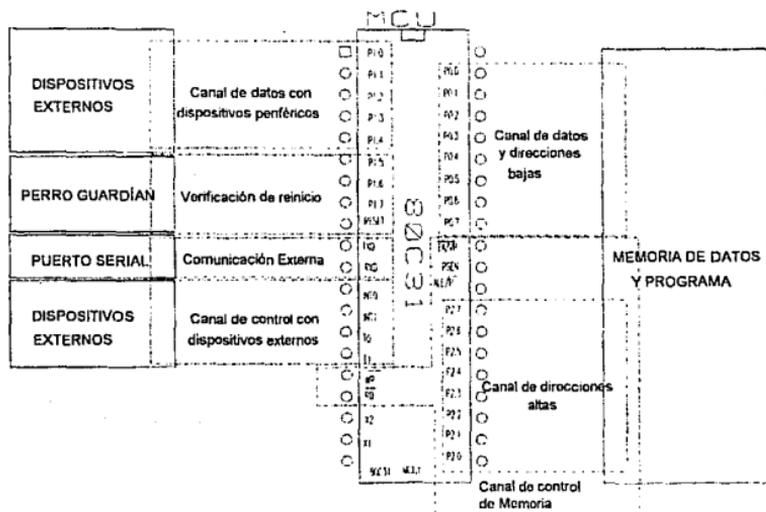
Para el diseño de las tarjetas de circuito impreso se adoptaron las siguientes convenciones:

1. Ancho de una línea de señal 0.015"
(0.381 mm.).
2. Ancho de una línea de alimentación (Vcc o GND)
0.025" (0.635 mm.).
3. Ancho de una conexión de señales 0.035"
(0.889 mm.).
4. Ancho de una conexión de alimentación 0.045"
(1.143 mm.).
5. Distancia mínima entre líneas
0.015" (0.381 mm.).
6. Distancia mínima entre líneas y conexiones
0.015" (0.381 mm.).
7. Distancia mínima entre conexiones 0.020"
(0.508 mm.).

En las tarjetas de sistema de procesamiento y de aplicación se ocupan ambas caras, con lo que se reduce la densidad de la líneas y la complejidad del circuito. La tarjeta de teclado es suficientemente sencilla para realizarla en una sola cara.

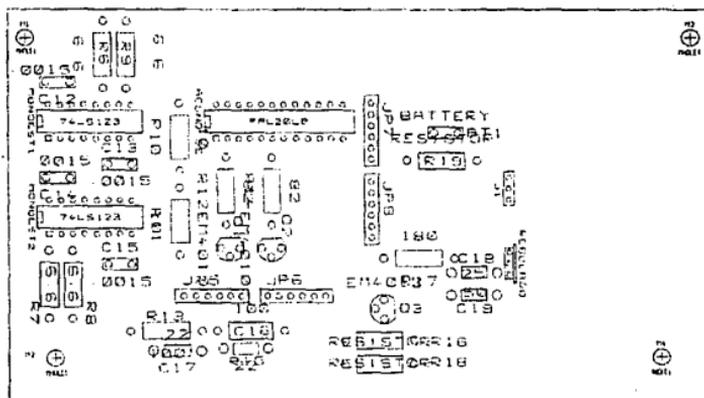
El diseño de la tarjeta de procesamiento se dividió en 3 partes: Memoria, comunicaciones y control. La ubicación de estas partes siguió la disposición de las señales en el microcontrolador, de acuerdo al esquema de la figura #63.

Fig.#63 Disposición de las terminales en el circuito microcontrolador



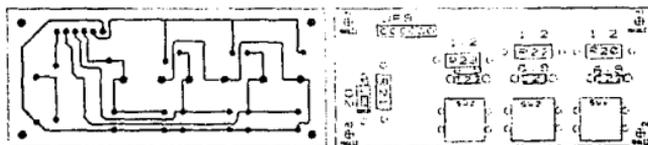
La mayor parte de las líneas de una cara tienen una sola dirección, de esta forma el circuito es mucho más sencillo de elaborar y optimizar. En la figura #64 se muestra el diagrama del circuito en forma general.

Fig.#65 Tarjeta de aplicación



Al terminar el diseño, las tarjetas de procesamiento y de aplicación fueron echas por la compañía JIN S.A., en fibra de vidrio, con esmalte antisoldadura. La tarjeta del teclado fue realizada en el propio Centro de Instrumentos, con técnicas de serigrafía y se ensambló de acuerdo al esquema de la figura #.66

Fig.#66 Teclado del sistema



Programación del PAL PARA EL DISPOSITIVO APUNTADOR
 ; JED2EQN -- JEDEC file to Boolean Equations disassembler
 (Version V015)
 ; Copyright (c) National Semiconductor Corporation 1990,1991
 ; Disassembled from E:\OPAL\MOUSE.JED. Date: 2-4-94

chip MOUSE PAL20L8

i1 i2 i3 i4 i5 i6 i7 i8 i9 i10 i11 GND
 i13 nc o15 nc f17 f18 o19 o20 o21 o22 nc VCC

equations

/o22 = /i7 + /i8
 o22.oe = vcc

/o21 = /i9 + /i10 + /i11
 o21.oe = vcc

/o20 = f18
 o20.oe = vcc

/o19 = f17
 o19.oe = vcc

/f18 = /i2 * /i6 + /i4 * i5 + /i1 * i6 + /i3 * /i5
 f18.oe = vcc

/f17 = /i4 * /i5 + /i2 * i6 + /i3 * i5 + /i1 * /i6
 f17.oe = vcc

/o15 = i13
 o15.oe = vcc

 Programación del PAL PARA DECODIFICACIÓN DE MEMORIA

; JED2EQN -- JEDEC file to Boolean Equations disassembler
(Version V015)

; Copyright (c) National Semiconductor Corporation 1990,1991

; Disassembled from E:\OPALMEMORIA.JED. Date: 2-4-94

chip MEMORIA PAL20L8

i1 i2 i3 i4 i5 i6 i7 i8 nc nc nc GND
nc nc o15 o16 o17 o18 o19 o20 o21 o22 nc VCC

equations

/o22 = /i4
o22.oe = vcc

/o21 = /i5 * /i6 * /i7 + /i5 * i6 * /i7
o21.oe = vcc

/o20 = /i3 * /i5 * /i6 * /i7 + /i1 * /i5 * i6 * /i7
o20.oe = vcc

/o19 = /i5 * /i6 * /i7
o19.oe = vcc

/o18 = /i2 * /i5 * /i6 * /i7 * /i8
o18.oe = vcc

/o17 = i5 * /i6 * /i7
o17.oe = vcc

/o16 = i2
o16.oe = vcc

/o15 = /i1 * /i5 * /i6 * /i7
o15.oe = vcc

E. Rutinas básicas.

El software del sistema de medición se encuentra dividido en 3 procedimientos principales: 1) adquisición y conversión de datos, 2) comunicación y transmisión de datos y 3) manipulación y edición de datos. Cada uno de ellos funciona independientemente, haciendo distintos usos de los registros de memoria. Estos procedimientos comparten el uso de una serie de subrutinas básicas, tales como: despliegue de mensajes, escritura de caracteres en la pantalla, verificación de operación al perro guardián etc. Se muestra la forma en que operan las siete rutinas básicas, ya que su funcionamiento es común para cualquier proceso.

1. Rutina Inicial.

La rutina inicial del sistema es responsable de poner en operación a todos los dispositivos y dar las condiciones iniciales de la operación del aparato, además de servir de acceso a todos los demás procedimientos del sistema. Sus funciones principales son:

- Activar la operación del perro guardián.
- Inicializar los primeros registros a usar.
- Inicializar la pantalla de despliegue.
- Presentar al usuario el menú de opciones de procedimientos.
- Correr el procedimiento elegido.

2. Inicialización de la pantalla de despliegue.

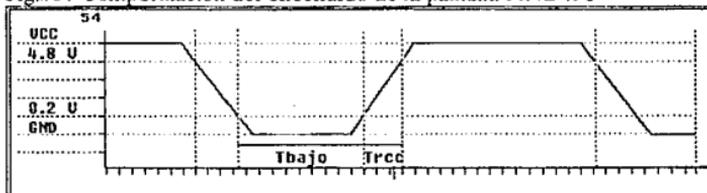
En la sección acerca de las rutinas de despliegue, se hará una explicación más detallada de la operación del despliegue. El objetivo de esta rutina es configurar al controlador de la pantalla para su funcionamiento adecuado, de acuerdo a las características del sistema. La pantalla de cristal líquido, AND 491, pertenece a una familia de despliegues que operan alrededor de la misma lógica de control y difieren exclusivamente en el tamaño (número de

caracteres), resolución (cantidad de puntos que forman un carácter) y arreglo (uno o dos renglones) con el que está formado la pantalla. Por lo tanto, es necesario inicializar al controlador de acuerdo al tipo de despliegue de que se trate. El controlador realiza una inicialización en el momento del encendido, que ejecuta las siguientes instrucciones:

- 1° Borra el contenido del despliegue.
- 2° Fija la longitud de la palabra de enlace con el microprocesador a 8 bits; fija el uso de caracteres de 5x7 puntos; y habilita el uso de una sola línea de despliegue.
- 3° Deshabilita al despliegue, al cursor y al parpadeo del cursor.
- 4° Coloca la pantalla en modo de entrada de datos, para que realice un incremento automático del contador de dirección cada vez que es transferido un dato. Además pone a la pantalla a funcionar sin sustitución de caracteres.
- 5° Es seleccionada la memoria de datos de despliegue para recibir la siguiente transmisión de datos.

Esta inicialización es genérica para cualquier tipo de pantalla de esta familia. Para que se garantice que la inicialización se ha realizado de forma correcta, se tienen que cumplir las condiciones de tiempos de encendido y apagado que muestra la figura #67 [12].

Fig.#67 Temporización del encendido de la pantalla AND491

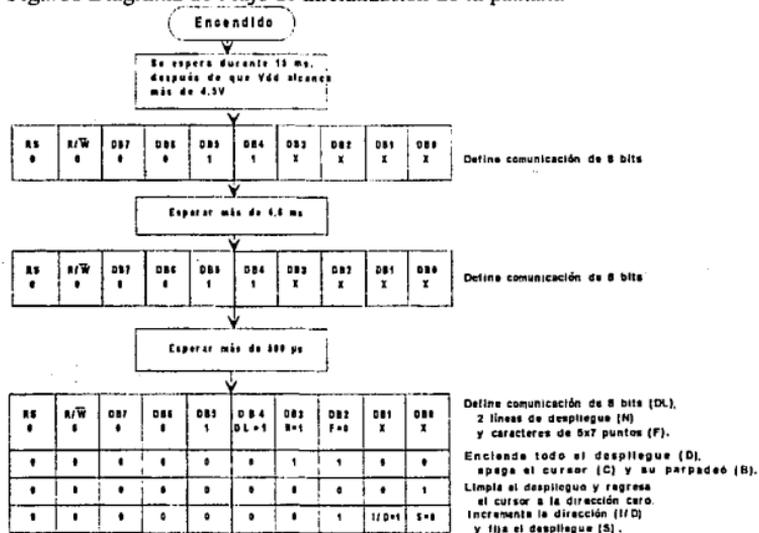


El lapso de duración de la transición de apagado a encendido debe ser de entre $0.1\text{ms} \leq T_{rcc} \leq 10\text{ms}$. y debe permanecer a menos

de 0.2 V durante por lo menos 1ms (Tbajo) antes de volver a encenderla. Si estas condiciones no se cumplen, la pantalla tiene que ser inicializada manualmente. Por otra parte, detalles como: número de renglones, incremento automático del apuntador de localidades, etc., tienen que ser ajustados por el microcontrolador. Entonces, con el fin de garantizar siempre la buena operación de la pantalla, se estableció que inicialización sería realizada por el microcontrolador, ajustándola a los requerimientos propios de operación.

La rutina de inicialización de la pantalla ajusta las condiciones de trabajo de acuerdo al diagrama de flujo mostrado en la fig. #68.

Fig.#68 Diagrama de Flujo de inicialización de la pantalla



3. Atención al Teclado.

Como se mencionó anteriormente, para la entrada de las distintas operaciones que es capaz de realizar el Ictiómetro se le

integró un teclado de 3 funciones; además de los dos botones de funciones con los que cuenta el ratón para la toma de datos. El funcionamiento del teclado puede ser atendido por poleo o por interrupciones. Sin embargo, para hacer uso de las facultades de detención de los procesos fue necesario hacer la implantación principal por medio de interrupciones; de esta forma se puede detener la operación del sistema hasta que se toque alguna tecla.

Estos teclados, aunque mantienen una similitud en funciones y operación, utilizan una señal de interrupción distinta. Para distinguirlos dentro de la tesis le llamaremos al primero teclado de comandos y al segundo teclado del ratón.

Al teclado de comandos le fue asignada la línea de interrupción 0. Cuando una tecla es accionada, se genera una señal de interrupción. La rutina de interrupción guarda en la pila el estado de algunos registros del sistema y selecciona un bloque de registro para trabajar. Primero, la rutina de atención al teclado verifica que tecla fue oprimida; lee el puerto y determina que señal se encuentra en un estado distinto. Después, el número de la tecla, (1, 2, o 3) es almacenado en una localidad de memoria interna (24H) y espera a que la tecla sea soltada. Por último, se restablece el estado que tenían los registros antes de ser generada la interrupción y se continúa el proceso. De esta forma el programa principal puede verificar que tecla fue accionada, leyendo esta localidad y realizar las operaciones adecuadas.

Para verificar que se ha dejado de accionar una tecla y evitar los reboles que se suceden cuando la tecla es accionada o liberada, se revisa el contenido de las señales del puerto; mientras haya alguna señal en bajo (significando que una tecla esta siendo accionada) se hace una espera y se procede a repetir el ciclo de verificación. Para la verificación, la rutina espera 5 ciclos más, aún cuando la tecla haya sido soltada, antes de regresar al proceso normal; si por alguna razón se vuelve a activar una tecla, la rutina reiniciará su operación de espera hasta que la tecla sea soltada.

4. Manejadores de Pantalla.

Antes de explicar las rutinas de operación de la pantalla hablaremos de la forma en que ésta opera internamente. Las pantallas de la familia de módulos de matriz de puntos AND son despliegues de cristal líquido de tipo alfanumérico, con una lógica de control integrada y con facilidades de interface con un procesador digital.

El controlador de la pantalla se enlaza a través de 2 registros, el Registro de Instrucciones (IR) y el Registro de Datos (DR). Por medio del registro IR se da entrada a los comandos, como pueden ser la limpieza de la pantalla o una dirección de la pantalla. Este registro es accesado solamente para escrituras (el MPU no puede leerlo). Cuando se hacen lecturas a este registro, el dato representa el valor del contador de direcciones (en los bits DB0 a DB6) y el valor de la bandera de estado "ocupado" (DB7).

El registro DR sirve como almacenamiento temporal para los datos que son enviados a la memoria de datos o a la de generación de caracteres. Una operación interna es la que envía los datos a cada memoria. A través de este registro también se realiza la lectura de la información almacenada, tanto en la memoria de datos como en la de generación de caracteres. Para acceder estos registros se hace una selección por medio de las señales RS (selección de registro) y R/W(lectura/escritura), de la pantalla de acuerdo a la siguiente tabla:

| RS | R/W | OPERACIÓN |
|----|-----|--|
| 0 | 0 | Escribe operaciones internas en el registro IR |
| 0 | 1 | Lectura de estado: Bandera de ocupado (DB7) y Contador de direcciones (DB0 a DB6). |
| 1 | 0 | Escribe al generador de caracteres o a la RAM del generador de caracteres |
| 1 | 1 | Lee el contenido de la RAM de despliegue o del generador de caracteres |

El registro de contador de dirección funciona como apuntador de la memoria, tanto a la de datos como a la de generación de caracteres. Este registro es accesado a través de IR. Cada vez que es escrito un dato en cualquiera de las memorias, este registro es incrementado (o decrementado de acuerdo a su configuración) en una localidad.

La memoria de datos de despliegue almacena los datos a desplegar en forma de códigos de 8 bits. La forma en que están compuestos dichos caracteres, así como los códigos que los representan, son mostrados en la figura #69[12].

Fig.#69

| Upper 4 bit Lower 4 bit | 0000 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|----------------------------|-----------------|------|------|------|------|------|------|------|------|------|------|------|------|
| XXXX0000 | CO HAL 11 | | Q | W | P | ' | E | - | 2 | E | O | P | |
| XXXX0001 | (2) | U | I | H | Q | a | a | . | P | F | L | a | g |
| XXXX0010 | (3) | " | Z | E | E | b | r | " | ' | W | X | P | E |
| XXXX0011 | (4) | # | B | O | S | c | e | L | ' | W | T | E | e |
| XXXX0100 | (5) | \$ | A | D | T | d | t | . | I | T | T | P | O |
| XXXX0101 | (6) | % | E | E | U | e | u | . | a | f | l | e | u |
| XXXX0110 | (7) | & | E | F | U | f | U | a | n | c | e | p | z |
| XXXX0111 | (8) | ' | Z | G | W | g | w | . | f | f | z | g | n |
| XXXX1000 | (1) | (| C | E | H | x | H | x | . | o | a | u | x |
| XXXX1001 | (2) |) | E | I | V | i | v | . | a | t | j | u | y |
| XXXX1010 | (3) | * | : | I | Z | j | z | . | e | o | o | l | f |
| XXXX1011 | (4) | + | : | K | O | k | o | . | a | a | o | * | a |
| XXXX1100 | (5) | , | < | L | f | l | f | . | a | a | f | w | a |
| XXXX1101 | (6) | - | = | M | O | m | o | . | u | z | z | e | + |
| XXXX1110 | (7) | . | > | H | ' | h | ' | . | a | e | h | ' | n |
| XXXX1111 | (8) | / | ? | O | _ | o | _ | . | e | w | u | z | o |

Como se puede observar, la mayor parte de los caracteres alfanuméricos corresponden al código ASCII, lo que facilita mucho el manejo de la pantalla.

Para el control de la pantalla existen 11 comandos que se presentan en la siguiente tabla:

| INSTRUCCIÓN | CÓDIGO DE LA INSTRUCCIÓN | | | | | | | | | |
|--|--------------------------|---------|-----------------|---------|---------|---------|---------|---------|---------|---------|
| | RS | R/ W | DB 7 | DB 6 | DB 5 | DB 4 | DB 3 | DB 2 | DB 1 | DB 0 |
| Limpia Despl. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Regresa a inicio | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | X |
| Modo de entrada | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S |
| Encendido del despl. | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | P |
| Mov. cursor y despl. | 0 | 0 | 0 | 0 | 0 | 1 | S/ C | R/L | X | X |
| Selección de funciones | 0 | 0 | 0 | 0 | 1 | DL | N | F | X | X |
| Selección de direc. de RAM | 0 | 0 | 0 | 1 | ACG | | | | | |
| Selección de direc. de RAM de despl. | 0 | 0 | 1 | ADD | | | | | | |
| Bandera de ocupado y dirección | 0 | 1 | BF | AC | | | | | | |
| Escribe datos a gen. de caract. o a la RAM de despliegue | 1 | 0 | Dato a escribir | | | | | | | |
| Lee datos del gen. de caract. o de la RAM de despliegue | 1 | 1 | Dato a leer | | | | | | | |

Donde:

I/D = Incrementar / decrementar la dirección de despliegue.

D = Encender o apagar el despliegue.

C = Desplegar el curso.

P = Parpadeo del cursor.

S/C = Indica si será el cursor o la pantalla la que se recorran cada vez que es ingresado un caracter.

R/L = Indica el sentido del corrimiento antes citado.

DL = Longitud de la palabra de comunicación: 4 ú 8 bits.

N = Número de líneas desplegadas.

F = Indica el tamaño de la letra; en el caso de nuestra pantalla solo pueden desplegarse letras de 5x7 puntos.

ACG = Apuntador de direcciones del generador de caracteres.

ADD = Apuntador de direcciones de datos.

BF = Bandera de ocupado.

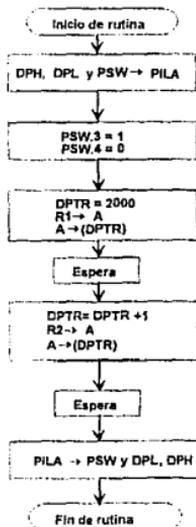
AC = Contador de direcciones.

La pantalla es manipulada por 3 rutinas principales: escritura de un carácter, escritura de un número hexadecimal y escritura de letrero. Estas rutinas fueron creadas con el fin de acceder la pantalla de manera sencilla desde cualquier parte del programa.

La rutina de escritura de carácter (fig. #70) tiene como función desplegar en la pantalla un carácter específico en una localidad determinada, requiriendo el código de este carácter y el código de la localidad que va a tener en la pantalla. Para su manipulación, el dato del carácter es almacenado en el registro 12H de la memoria interna por la rutina que desee desplegarlo; mientras tanto, la dirección es almacenada en el registro 11H. Después de almacenarlos, es hecho el llamado a la subrutina de despliegue.

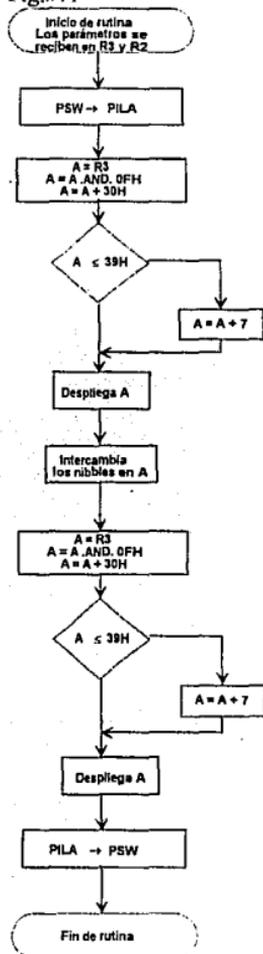
Primero, la rutina se encarga de enviar la localidad en la pantalla a la dirección de memoria

Fig.#70 Rutina de escritura de carácter a la pantalla



DPH = Parte alta del apuntador de datos
 DPL = Parte baja del apuntador de datos
 PSW = Registro de banderas
 A = Acumulador
 DPTR = Apuntador de datos (16 bits)

Fig.#71



externa; donde se encuentra el registro de instrucciones de la pantalla (2000H). Se ejecuta entonces una rutina de espera, mientras la pantalla realiza las operaciones internas, para en seguida enviar al registro de datos (DR) el código del carácter. Como todas las rutinas, esta salva en la pila el contenido de los registros que utiliza y al finalizar los restaura.

La rutina de escritura de un número hexadecimal tiene la finalidad de desplegar una variable en su valor hexadecimal, como puede ser el contenido de un registro interno. La rutina recibe como variables al valor a desplegar en la dirección 13H de memoria interna y la localidad de despliegue en la dirección 11H.

La rutina hace la transformación del valor hexadecimal al par de caracteres ASCII, que representan la parte baja y la alta del valor. Esta transformación se realiza de acuerdo al diagrama de flujo de la figura #71. Cada vez que la rutina ha obtenido el carácter ASCII correspondiente, la información es enviada a la pantalla haciendo uso de la rutina de despliegue de carácter.

Por último, la rutina de escritura de letrero se desarrolló para facilitar el despliegue de una secuencia de caracteres, como puede ser un mensaje al operador, de tal forma que el cálculo de localidades de la

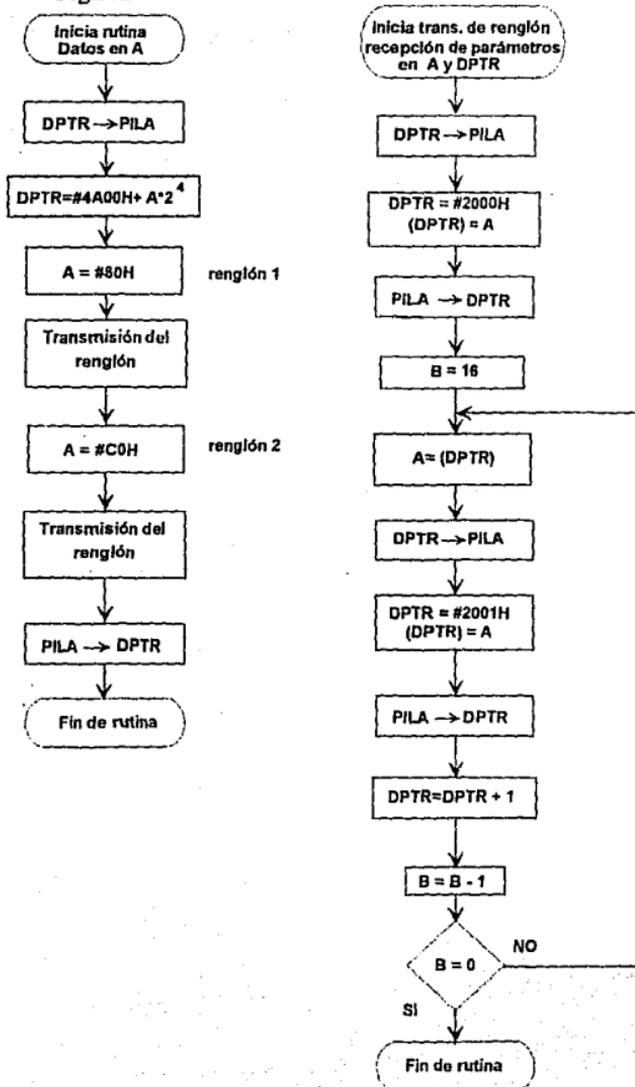
pantalla fuese más rápido. Los mensajes están almacenados en la memoria EPROM, por lo que la rutina solo requiere saber cual de los letreros se desea desplegar.

La rutina se divide en dos partes, la primera se encarga de hacer los cálculos de direcciones y la segunda se encarga de las transmisiones (ver la fig. #72).

En la primera parte, se recibe el número de letrero en la dirección 27H de memoria interna; por medio de la adición de un valor de referencia se obtiene la dirección de memoria externa en la que se encuentran almacenados los caracteres. Con las direcciones ya calculadas se realiza la parte de transmisión. Ahí, los caracteres son capturados, se calcula la localidad en la pantalla donde se deben de ubicar y se transmiten. Es necesario hacer uso de la pila para poder intercambiar la dirección con facilidad ya que la memoria EPROM (donde se encuentran los datos) y la pantalla se encuentran ubicada en el mapa de memoria de datos. Se hace una primera transmisión de los caracteres que pertenecen al primer renglón y se regresa a la primera rutina. Luego direcciona el segundo renglón y se realiza el envío.

La rutina también se encarga de las esperas necesarias para la sincronización con la pantalla.

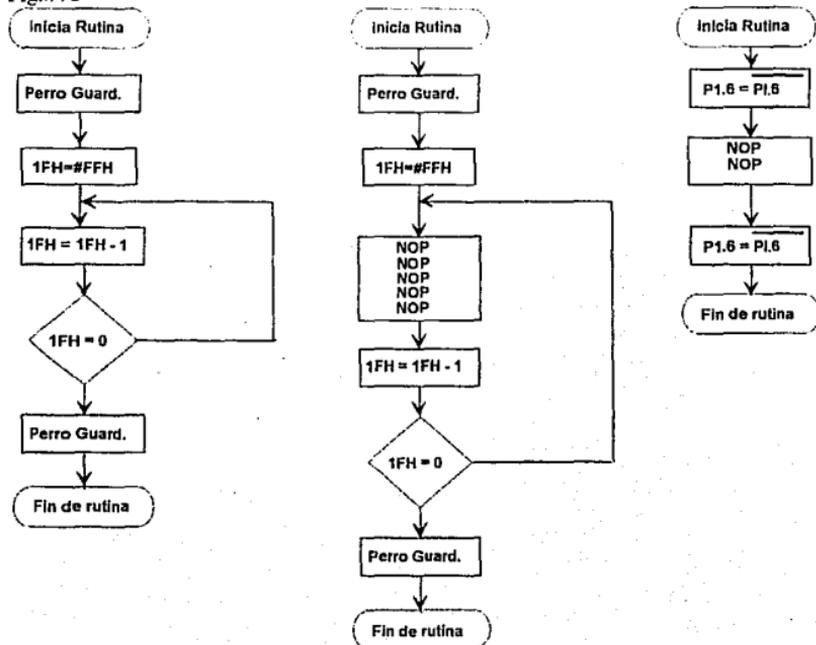
Fig.#72



5. Rutinas de espera.

El sistema requiere de rutinas que suspendan la operación mientras se realizan procesos por dispositivos externos o mientras se cumple una cierta condición: Es necesario hacer que el microcontrolador no realice operación alguna hasta que no haya terminado la causa de la demora. Estas rutinas son importantes para la sincronización de todo el sistema. Se realizaron dos rutinas con dos magnitudes distintas de retraso (fig. #73).

Fig.#73



La primera rutina, de retraso corto, realiza 255 veces la instrucción de "decrementa y salta si no es cero" (DJNZ); esta instrucción toma 24 periodos de oscilación para ejecutarse. Como la frecuencia de oscilación en nuestro sistema es de 12 Mhz., entonces hablamos de un tiempo de retraso de:

$$\frac{24 \left(\frac{\text{OSC.}}{\text{CICL.}} \right) \times 255 \left(\frac{\text{CICL.}}{\text{RUT.}} \right)}{12 \times 10^6 \left(\frac{\text{OSC.}}{\text{SEG.}} \right)} = \frac{6120 \left(\frac{\text{OSC.}}{\text{RUT.}} \right)}{12 \times 10^6 \left(\frac{\text{OSC.}}{\text{SEG.}} \right)} = 510 (\mu\text{s.})$$

Este es un valor aproximado, ya que para el obtener el valor exacto, se le tiene que agregar los tiempos ocupados en el salto a la subrutina y la inicialización de variables. Además, cada vez que se ejecutada la rutina, se realiza una verificación al "perro guardián" antes y después del retardo. Todas estas operaciones afectan el tiempo real de espera. En total se realizan: 2 llamadas a rutinas (ACALL) de 24 periodos de oscilación cada una, 4 operaciones de complemento de bit (CLP, 12 periodos), 4 "no operación" (NOP, 12 periodos), 3 "regreso de rutina" (RET, 24 periodos) y 1 movimiento de dato (MOV, 12 periodos). Esto significa un retardo adicional de 228 periodos de oscilación por rutina, o sea 19 μs . Si añadimos este tiempo al de nuestra rutina obtenemos un retraso aproximado de 529 μs .

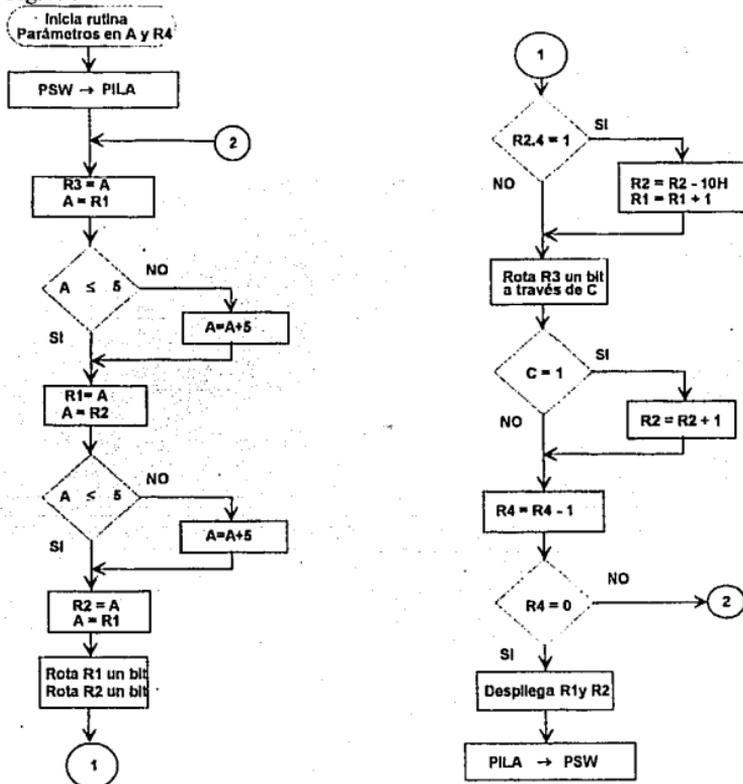
Existe otra rutina de mayor duración, que inserta más instrucciones de "no operación" (NOP); por lo que se realizan 4 operaciones más. En total se realizan 48 periodos por cada uno de los 255 ciclos que realiza la rutina o sea 1020 μs . además de los 529 μs . ya calculados. Estas rutinas pueden ser usadas múltiples veces para alcanzar retardos aún mayores.

6. Conversión binario - decimal.

Una vez realizada las operaciones, se procede a hacer el despliegue de los resultados. Para hacer el despliegue de datos se

realiza una conversión hexadecimal a BCD (fig. #74) y luego su caracter ASCII. Esta rutina se basa en la conversión estándar BCD - hexadecimal. La rutina opera haciendo un corrimiento a la izquierda del valor, o sea, se obtienen las potencias de 2 del número.

Fig. #74



Si se hace un corrimiento a la izquierda de un valor igual o mayor a 5, el resultado en BCD será mayor a 10 (1-0: 2 dígitos); el valor de las decenas debe pasar a otro registro, pero como se trata de un valor binario este se almacenaría como 0AH en un solo byte, lo cual ocasiona un error. Para evitarlo, cuando se va a realizar el corrimiento se verifica si es un valor superior o igual a 5. De serlo, se le suma un valor de 3 (debido a que se debe de ajustar la diferencia de las bases 16 y 10) para que en el siguiente corrimiento se obtenga precisamente el valor 1xH. El ciclo se repite 8 veces, una para cada bit del registro.

Se procede entonces a la transmisión de los datos a la pantalla. Para realizar el despliegue de las fracciones primero debemos transformar el valor de la cuenta de fracciones de pulgada. Para representar su valor en BCD, cada paso representa 0.005 (1/200) de fracción de pulgada recorrida y es representado por una cuenta de 01H (0000 0001B). Se observa, en la siguiente tabla, la relación entre los valores en binario y en fracciones de pulgada:

| PASOS | HEXADECIMAL | BINARIO | PULGADAS |
|-------|-------------|------------|----------|
| 1 | 01H | 0000 0001B | .005 |
| 45 | 2DH | 0010 1101B | .225 |
| 172 | 48H | 0100 1000B | .860 |

De las cantidades en binario se puede deducir que el bit menos significativo de la cuenta nos representa cuando el valor menos significativo de las fracciones de pulgada, es 0.005 o 0. Mientras que los 7 bits más significativos de las fracciones representan el valor de las décimas y centésimas de la fracción. Por ejemplo, si del el valor de 45 pasos tomamos solamente los 7 bits más significativos tenemos 0001 0110B; si hacemos la conversión HEXADECIMAL - BCD obtenemos el número 22D. Por este motivo, se procede ha

realizar la conversión hexadecimal - BCD únicamente de los 7 bits más significativos y se usa al 8 bit para determinar si se despliega un valor de 0 o 5 como último dígito.

F. Listado de programas.

```

.....
;
;           VECTORES DE INTERRUPCION
; En esta zona direcciona las rutinas de atención a las interrupciones
;
;
.....
0000          ORG 0000H          ;*****RESET*****
0000 0191     ETIQ: AJMP ETIQ0
0003          ORG 0003H          ;*****Vector Interup Ex1*****
0003 0203CD   LJMP RIE0
000B          ORG 000bh          ;*****Vector interup Tx0*****
000B 0503     INC 003H
000D 32       RETI
001B          ORG 001bh          ;*****Vector interup Tx1*****
001B 0500     INC 000H
001D 32       RETI
0013          ORG 0013H          ;*****Vector interup EX0*****
0013 0130     AJMP ETIQ1 ; NO Hay espacio,interrupciones llegan
hasta la 02Bh
.....
;
;           RUTINA DE ATENCION A LA INTERRUPCION EXTERNA #1
; USADA EN EL MANEJO DE LOS BOTONES DEL DISPOSITIVO APUNTAADOR
;
;
.....
0030          ORG 0030H
0030 C0D0     ETIQ1: PUSH PSW
0032 C011     PUSH 011H
0034 C012     PUSH 012H
0036 C013     PUSH 013H

```

```

0038 C0A8          PUSH IE
003A C0E0          PUSH 0E0H
003C C2AF          CLR IE.7
003E C2D4          CLR PSW.4
0040 C2D3          CLR PSW.3
0042 C006          PUSH R6
0044 209205        JB P1.2,ETIQ11      ; P1.2
0047 752403        MOV 024H,#003H     ; TECLA 3
004A 0157          AJMP ETIQ2
004C 209305        ETIQ11: JB P1.3,ETIQ12 ; P1.3
004F 752402        MOV 024H,#002h    ; TECLA 2
0052 0157          AJMP ETIQ2
0054 752401        ETIQ12: MOV 024H,#001h ; Tecla 1
0057 7E05          ETIQ2: MOV R6,#005H ; Espera hasta soltar tecla
0059 118A          ESPER1: ACALL PERRO
005B E590          MOV A,P1
005D 541C          ANL A,#01CH
005F B41CF5        CJNE A,#01CH,ETIQ2
0062 BE0002        CJNE R6,#000H,ESPER3
0065 8005          SJMP REGINT
0067 1E           ESPER3: DEC R6
0068 117B          ACALL ESPE
006A 80ED          SJMP ESPER1
006C D006          REGINT: POP R6
006E D0E0          POP 0E0H
0070 D0A8          POP IE
0072 D013          POP 013H
0074 D012          POP 012H
0076 D011          POP 011H
0078 D0D0          POP PSW
007A 32           RETI

```

.....
; RUTINA DE ESPERA

; Con el fin de sincronizar el microprocesador con otros procesos,
; la rutinas que se encargaran de ocupar el procesador.

```

.....
;
007B 118A     ESPE: ACALL PERRO
007D 751FFF           MOV 01FH,#0FFH
0080 00         ESP:  NOP
0081 00         NOP
0082 00         NOP
0083 00         NOP
0084 D51FF9           DJNZ 01FH,ESP
0087 118A         ACALL PERRO
0089 22         RET
.....

```

```

; HABILITA WATCH-DOG

```

```

; Esta rutina se encarga de reportar la actividad normal al perro guardián
; en caso de no ejecutarse periódicamente el perro guardián reiniciará
; el sistema
.....

```

```

008A B296     PERRO: CPL P1.6
008C 00         NOP
008D 00         NOP
008E B296           CPL P1.6
0090 22         RET
.....

```

```

; PROGRAMA PRINCIPAL

```

```

; Esta rutina es cargada al iniciar la operación del dispositivo, y despliega
; las opciones de procesos que podemos llevar a cabo.
;
.....

```

```

0091 758128     ETIQ0: MOV SP,#028H           ; Inicia stack
0094 118A         ACALL PERRO
0096 719D         ACALL INDSP
0098 C2D4         CLR PSW.4           ; Define banco de registros 1
009A D2D3         SETB PSW.3
009C 7400         MOV A,#000H           ; Despliega el primer letrero
009E 12093C       LCALL LDSP ;+++++++PRESENTACION+++++++
00A1 D2AF     ETIQ0R: SETB IE.7           ; Habilita toda interrupcion

```

```

00A3 D2AA      SETB IE.2      ; Habilita Int. Externa 1
00A5 D28A      SETB TCON.2    ; Int.Ext.1 "falling edge"
00A7 7401      MOV A,#001H    ; Despliega el segundo letrero
00A9 120956    LCALL IMPSP ;++++OPCIONES DE PROCESOS
00AC 14        DEC A
00AD 23        RL A
00AE 23        RL A
00AF 9000B3    MOV DPTR,#(CASE1)H ;Formulación del salto tipo
CASE
00B2 73        JMP @A+DPTR
00B3 D150      CASE1: ACALL DATOS
00B5 8006      SJMP YREG
00B7 F11B      ACALL SCOM
00B9 8002      SJMP YREG
00BB 11C5      ACALL EDIT
00BD 7FFF      YREG: MOV R7,#0FFH
00BF 117B      ETIQ00: ACALL ESPE
00C1 DFFC      DJNZ R7,ETIQ00
00C3 80DC      SJMP ETIQ0R
;.....
; PROCEDIMIENTO DE EDICION
; Este permite borrar la memoria por completo, eliminar algún dato o
; corregirlo
;.....
00C5 900000    EDIT: MOV DPTR,#0000
00C8 E0        MOVX A,@DPTR
00C9 6006      JZ MEMO
00CB 740A      MOV A,#80AH
00CD 12093C    LCALL LDSP;++++NO HAY DATOS QUE
EDITAR++++
00D0 22        RET
00D1 740B      MEMO: MOV A,#00BH
00D3 120956    LCALL IMPSP ;++++OPCIONES DE
EDICION++++
00D6 14        DEC A

```

```

00D7 23          RL A
00D8 23          RL A
00D9 9000DD     MOV DPTR,#(CASE2)H ; Formulación del salto tipo
CASE
00DC 73          JMP @A+DPTR
00DD 11E8       CASE2: ACALL BORRA
00DF 8006       SJMP RTM
00E1 312B       ACALL BARCH
00E3 8002       SJMP RTM
00E5 31F3       ACALL EDATO
00E7 22         RTM: RET
;.....
; RUTINA DE BORRADO
; Se encarga de reiniciar todos los valores de la memoria cambiando su
; contenido por FFh
;
;.....
00E8 7409       BORRA: MOV A,#009H
00EA 120956     LCALL IMPSP ;++++CONFIRMA BORRADO+++++
00ED B4011E     CJNE A,#001H,SALR
00F0 900000     MOV DPTR,#0000H
00F3 7403       MOV A,#003H ; Imprime letrero
00F5 120918     LCALL LTR ;++++BORRANDO MEMORIA ESPERE..
00F8 74FF       BORR: MOV A,#0FFH ; Borra toda la memoria (ciclo iterativo)
00FA F0         MOVX @DPTR,A
00FB A3         INC DPTR
00FC E583       MOV A,DPH
00FE B41F08     CJNE A,#01FH,NP
0101 7404       MOV A,#004H ; Imprime mensaje de terminación
0103 12093C     LCALL LDSP ;+++++TERMINO DE
BORRAR+++++++
0106 22         RET
0107 117B       NP: ACALL ESPE ; Espera que se haya grabado el dato
0109 3097FB     JNB P1.7,NP
010C 80EA       SJMP BORR
010E 22         SALR: RET

```

```

010F E51A      MARCH: MOV A,01AH
0111 B50805    CJNE A,008H,AUMT
0114 751A01    MOV 01AH,#001H
0117 801B      SJMP ESPLM
0119 051A      AUMT: INC 01AH
011B 8017      SJMP ESPLM
011D E51A      MENARC: MOV A,01AH
011F B40105    CJNE A,#001H,DENP
0122 85081A    MOV 01AH,008H
0125 800D      SJMP ESPLM
0127 151A      DENP: DEC 01AH
0129 8009      SJMP ESPLM
012B 900001    BARCH: MOV DPTR,#0001H
012E E0        MOVX A,@DPTR
012F F508      MOV 008H,A
0131 751A01    MOV 01AH,#001H
0134 740D      ESPLM: MOV A,#00DH
0136 120918    LCALL LTR ;ESCOJE ARCH 1)MAS 2)MENOS
3)BORRA
0139 750C08    MOV 00CH,#008H
013C 750FCA    MOV 00FH,#0CAH
013F E51A      MOV A,01AH
0141 B199      ACALL HEXBCD
0143 120958    LCALL IMPS ; Imprime un n-mero hexadecimal sin
cambiar lo ya Impreso
0146 14        DEC A
0147 23        RL A
0148 90014C    MOV DPTR,#(CASE4)H ; Formulación del salto tipo
CASE 014B 73    JMP @A+DPTR
014C 80C1      CASE4: SJMP MARCH
014E 80CD      SJMP MENARC
0150 7409      MOV A,#009H
0152 120956    LCALL IMPSP ;CONFIRMA OPERACION en caso de
ser el botón 1
0155 B40164    CJNE A,#001H,RTPT
0158 900001    MOV DPTR,#001H

```

| | |
|-------------|-----------------------|
| 015B E508 | MOV A,008H |
| 015D B51A02 | CJNE A,01AH,CMPDAT |
| 0160 8054 | SJMP RNUM |
| 0162 851A08 | CMPDAT: MOV 008H,01AH |
| 0165 1508 | DEC 008H |
| 0167 7180 | ACALL ADDAT |
| 0169 7169 | ACALL CNNAD |
| 016B A3 | INC DPTR |
| 016C A3 | INC DPTR |
| 016D 858210 | MOV 010H,DPL |
| 0170 858311 | MOV 011H,DPH |
| 0173 900001 | MOV DPTR,#0001H |
| 0178 E0 | MOVX A,@DPTR |
| 0177 F508 | MOV 008H,A |
| 0179 C01B | PUSH 01BH |
| 017B C01C | PUSH 01CH |
| 017D 7180 | ACALL ADDAT |
| 017F 7169 | ACALL CNNAD |
| 0181 A3 | INC DPTR |
| 0182 A3 | INC DPTR |
| 0183 858212 | MOV 012H,DPL |
| 0186 858313 | MOV 013H,DPH |
| 0189 851A08 | MOV 008H,01AH |
| 018C 900001 | MOV DPTR,#0001H |
| 018F 7180 | ACALL ADDAT |
| 0191 7169 | ACALL CNNAD |
| 0193 A3 | INC DPTR |
| 0194 D01C | POP 01CH |
| 0196 D01B | POP 01BH |
| 0198 31C2 | ACALL BPART |
| 019A 751100 | MOV 011H,#000H |
| 019D E51A | MOV A,01AH |
| 019F 23 | RL A |
| 01A0 F510 | MOV 010H,A |
| 01A2 900001 | MOV DPTR,#0001H |
| 01A5 E0 | MOVX A,@DPTR |

```

01A6 23          RL A
01A7 2402       ADD A,#002H
01A9 F512       MOV 012H,A
01AB 751300     MOV 013H,#000H
01AE E51A       MOV A,01AH
01B0 23          RL A
01B1 04          INC A
01B2 F582       MOV DPL,A
01B4 31C2       ACALL BPART
01B6 900001     RNUM: MOV DPTR,#0001H
01B9 E0          MOVX A,@DPTR
01BA 14          DEC A
01BB F0          MOVX @DPTR,A
01BC 117B       RTPT: ACALL ESPE
01BE 3097FB     JNB P1.7,RTPT
01C1 22          RET

```

.....
; SUBROUTINA DE BORRADO DE UNA SECCION DE MEMORIA

; En caso de que se desee eliminar una sección de la memoria, es necesario
; reacomodar los datos para poder aprovechar al máximo su capacidad.

```

;.....
01C2 C2AF       BPART: CLR IE.7
01C4 740E       MOV A,#00EH
01C6 120918     LCALL LTR ;+++++COMPRESION DE MEMORIA+++++
01C9 A3         CIC:  INC DPTR
01CA E0         MOVX A,@DPTR
01CB C082       PUSH DPL
01CD C083       PUSH DPH
01CF 851082     MOV DPL,010H
01D2 851183     MOV DPH,011H
01D5 F0         MOVX @DPTR,A
01D6 117B       NP2: ACALL ESPE ; Espera que se haya grabado el dato
01D8 3097FB     JNB P1.7,NP2
01DB A3         INC DPTR
01DC 858210     MOV 010H,DPL

```

```

01DF 858311      MOV 011H,DPH
01E2 D083       POP DPH
01E4 D082       POP DPL
01E6 E583       MOV A,DPH
01E8 B513DE     CJNE A,013H,CIC
01EB E582       MOV A,DPL
01ED B512D9     CJNE A,012H,CIC
01F0 740E       MOV A,#00EH
01F2 22         RET

```

```

; SUBROUTINA DE EDICION DE DATOS

```

```

; En esta rutina se puede eliminar o modificar un dato en específico
;
;

```

```

01F3 C2AF      EDATO: CLR IE.7

```

```

01F5 752504    MOV 025H,#004H      ; Inicia variables

```

```

01F8 758A38    MOV TL0,#038H

```

```

01FB 758B38    MOV TL1,#038H

```

```

01FE 752000    MOV 020H,#000H

```

```

0201 C2D4      CLR PSW.4

```

```

0203 C2D3      CLR PSW.3

```

```

0205 7800      MOV R0,#000H

```

```

0207 7C00      MOV R4,#000H

```

```

0209 7D00      MOV R5,#000H

```

```

020B 7F00      MOV R7,#000H

```

```

020D 75A884    MOV IE,#084H

```

```

0210 75B805    MOV IP,#005H

```

```

0213 751801    MOV 018H,#001H

```

```

0216 751900    MOV 019H,#000H

```

```

0219 751A01    MOV 01AH,#001H

```

```

021C 900001    MOV DPTR,#0001H

```

```

;Carga número de archivos

```

```

021F 751B00    MOV 01BH,#000H

```

```

0222 751C00    MOV 01CH,#000H

```

```

0225 751D00    MOV 01DH,#000H

```

```

0228 751E00    MOV 01EH,#000H

```

```
022B 752800      MOV 026H,#000H
022E 51A4        ACALL CADDAT
0230 740C        MOV A,#00CH
0232 120918      LCALL LTR ;+++++++MASCARA DE EDICION++++
0235 7169      CICL: ACALL CNNAD
0237 A3          INC DPTR
0238 A3          INC DPTR
0239 E0          MOVX A,@DPTR      ; Lee el dato actual fraccion
023A 750FCB      MOV 00FH,#0CBH
023D 750C07      MOV 00CH,#007H
0240 B199        ACALL HEXBCD
0242 A3          INC DPTR
0243 E0          MOVX A,@DPTR      ; Lee el dato actual entero
0244 750FC8      MOV 00FH,#0C8H
0247 750C08      MOV 00CH,#008H
024A B199        ACALL HEXBCD
024C 751184      MOV 011H,#084H
024F 851A13      MOV 013H,01AH
0252 91DB        ACALL DHEX
0254 75118A      MOV 011H,#08AH
0257 851813      MOV 013H,018H
025A 91DB        ACALL DHEX
025C 751188      MOV 011H,#088H
025F 851913      MOV 013H,019H
0262 91DB        ACALL DHEX
0264 120958      LCALL IMPS
0267 14          DEC A
0268 23          RL A
0269 23          RL A
026A 90026E      MOV DPTR,#(CASE5)H ; Formulación del salto tipo
CASE 026D 73      JMP @A+DPTR
026E 5187      CASE5: ACALL MASAR
0270 8007        SJMP REGM
0272 51C0        ACALL MASDA
0274 8003        SJMP REGM
0276 51F8        ACALL EDITS
```

```

0278 22          RET
0279 751FFF     REGM: MOV 01FH,#0FFH
027C C01F      PTIM: PUSH 01FH
027E 117B      ACALL ESPE
0280 D01F      POP 01FH
0282 D51FF7     DJNZ 01FH,PTIM
0285 80AE      SJMP CICL

```

```

.....
; SUBROUTINA DE MANEJO DE ARCHIVO

```

```

; Esta rutina se encarga de adquirir los datos de un archivo en específico
; para su posible eliminación
;
;
.....

```

```

0287 751B00     MASAR: MOV 01BH,#000H
028A 751C0C     MOV 01CH,#000H
028D 751801     MOV 018H,#001H
0290 751900     MOV 019H,#000H
0293 900001     MOV DPTR,#0001H
0296 E0         MOVX A,@DPTR
0297 B51A03     CJNE A,01AH,PAUM
029A 751A00     MOV 01AH,#000H
029D 851A08     PAUM: MOV 008H,01AH
02A0 7180       ACALL ADDAT ; Llama a la rutina que determina la
cantidad de datos
02A2 051A       INC 01AH
02A4 E51A       CADDAT: MOV A,01AH
02A6 23        RL A
02A7 F582       MOV DPL,A
02A9 C3        CLR C
02AA E0        MOVX A,@DPTR
02AB 34FF       ADDC A,#0FFH ; Obtencion de la cantidad de datos
menos 1 por archivo
02AD F51D       MOV 01DH,A
02AF A3        INC DPTR
02B0 E0        MOVX A,@DPTR
02B1 F5F0       MOV B,A

```

```
02B3 541F      ANL A,#01FH
02B5 34FF      ADDC A,#0FFH
02B7 F51E      MOV 01EH,A
02B9 E5F0      MOV A,B
02BB 54E0      ANL A,#0E0H
02BD F526      MOV 026H,A
02BF 22        RET
```

.....
; SUBROUTINA DE MANEJO DE DATOS

; En esta rutina se obtiene los datos específicos de un dato para su posible
; modificación o eliminación

.....
02C0 E51D MASDA: MOV A,01DH
02C2 C3 CLR C
02C3 3401 ADDC A,#001H
02C5 B51819 CJNE A,018H,NL11
02C8 E51E MOV A,01EH
02CA 5001 JNC NCRY
02CC 04 INC A
02CD B51911 NCRY: CJNE A,019H,NL11
02D0 751801 MOV 018H,#001H
02D3 751900 MOV 019H,#000H
02D6 851A08 MOV 008H,01AH
02D9 1508 DEC 008H
02DB 900001 MOV DPTR,#0001H
02DE 7180 ACALL ADDAT
02E0 22 RET
02E1 E518 NL11: MOV A,018H
02E3 C3 CLR C
02E4 3401 ADDC A,#001H
02E6 5002 JNC NOCA
02E8 0519 INC 019H
02EA F518 NOCA: MOV 018H,A
02EC E51B MOV A,01BH
02EE C3 CLR C

```

02EF 3401      ADDC A,#001H
02F1 5002      JNC NHCA
02F3 051C      INC 01CH
02F5 F51B      NHCA: MOV 01BH,A
02F7 22        RET
02F8 740F      EDITS: MOV A,#00FH
02FA 120956    LCALL IMPSP ;+++++OPCION DE BORRAR O
EDITAR DATO
02FD 14        DEC A
02FE 23        RL A
02FF 23        RL A
0300 900304    MOV DPTR,#(CASE3)H ; Formulación del salto tipo
CASE 0303 73   JMP @A+DPTR
0304 7125      CASE3: ACALL BDAT
0308 8004      SJMP REGMZ
0308 710D      ACALL EDAT
030A 8000      SJMP REGMZ
030C 22        REGMZ: RET

```

```

; SUBROUTINA DE EDICION DE DATOS

```

```

; En caso de querer modificar un dato en especifico esta rutina proporciona
; un enlace con la rutina de adquisición de medidas

```

```

;
;
;
030D 851B21    EDAT: MOV 021H,01BH
0310 758805    MOV TCON,#005H
0313 75A88F    MOV IE,#08FH
0316 7405      MOV A,#005H
0318 120918    LCALL LTR
031B B1E7      SCLO: ACALL DESCU
031D E521      MOV A,021H
031F B51B02    CJNE A,01BH,REW
0322 80F7      SJMP SCLO
0324 22        REW: RET

```

```

; SUBROUTINA DE BORRAR DATO

```

; Se encarga de manejar el borrado de un dato, la modificación de los datos
; en la memoria y su reorganización

;

.....

| | | |
|-------------|-------------------|---|
| 0325 7169 | BDAT: ACALL CNNAD | |
| 0327 A3 | INC DPTR | |
| 0328 A3 | INC DPTR | |
| 0329 858210 | MOV 010H,DPL | ; Guarda la dirección en memoria |
| 032C 858311 | MOV 011H,DPH | |
| 032F 900001 | MOV DPTR,#0001H | |
| 0332 E0 | MOVX A,@DPTR | ; Determina la dirección hasta la que debe de modificar |
| 0333 F508 | MOV 008H,A | |
| 0335 C01B | PUSH 01BH | |
| 0337 C01C | PUSH 01CH | |
| 0339 7180 | ACALL ADDAT | |
| 033B 7169 | ACALL CNNAD | |
| 033D A3 | INC DPTR | |
| 033E A3 | INC DPTR | |
| 033F 858212 | MOV 012H,DPL | |
| 0342 858313 | MOV 013H,DPH | |
| 0345 D01C | POP 01CH | |
| 0347 D01B | POP 01BH | |
| 0349 851082 | MOV DPL,010H | |
| 034C 851183 | MOV DPH,011H | |
| 034F A3 | INC DPTR | |
| 0350 31C2 | ACALL BPART | |
| 0352 E51A | FIGU: MOV A,01AH | |
| 0354 23 | RL A | |
| 0355 758300 | MOV DPH,#000H | |
| 0358 F582 | MOV DPL,A | |
| 035A E51D | MOV A,01DH | |
| 035C F0 | MOVX @DPTR,A | |
| 035D E51C | MOV A,01CH | |
| 035F 4526 | ORL A,026H | |

```

0361 A3          INC DPTR
0362 F0          MOVX @DPTR,A
0363 117B        RETRA: ACALL ESPE
0365 3087FB     JNB P1.7,RETRA
0368 22          RET
0369 C3          CNNAD: CLR C          ; Dirección de datos
036A E51B        MOV A,01BH
036C 33          RLC A
036D F582        MOV DPL,A
036F E51C        MOV A,01CH
0371 33          RLC A
0372 F583        MOV DPH,A
0374 C3          CLR C
0375 E582        MOV A,DPL
0377 34BE        ADDC A,#0BEH
0379 F582        MOV DPL,A
037B 5002        JNC MRGY
037D 0583        INC DPH
037F 22          MRGY: RET

```

```

.....
; LECTURA DE CANTIDAD DE DATOS

```

```

; Esta rutina se encarga de obtener a partir de un numero de archivo cuantos
; datos existen en la memoria
;

```

```

.....
0380 751B00     ADDAT: MOV 01BH,#000H
0383 751C00     MOV 01CH,#000H
0388 E508       MOV A,008H
0388 6012       JZ SASY
038A A3         ADDAT1: INC DPTR
038B E0         MOVX A,@DPTR
038C C3         CLR C
038D 351B       ADDC A,01BH
038F F51B       MOV 01BH,A
0391 A3         INC DPTR
0392 E0         MOVX A,@DPTR

```

0393 541F ANL A,#01FH ; Elimina el tipo de archivo y almacena solo la cantidad.

0395 351C ADDC A,01CH

0397 F51C MOV 01CH,A

0399 D508EE DJNZ 008H,ADDAT1

039C 22 SASY: RET

.....
; RUTINA DE INICIALIZACION DE DESPLIEGUE

; Esta rutina configura la pantalla de LCD para su operación

.....
039D 7B03 INDSP: MOV R3,#003H

039F 117B CL2: ACALL ESPE

03A1 DBFC DJNZ R3,CL2

03A3 902000 MOV DPTR,#2000H

03A6 743F MOV A,#03FH

03A8 F0 MOVX @DPTR,A

03A9 7B08 MOV R3,#008H

03AB 117B CLA2: ACALL ESPE

03AD DBFC DJNZ R3,CLA2

03AF F0 MOVX @DPTR,A

03B0 117B ACALL ESPE

03B2 7900 MOV R1,#000H

03B4 743B MOV A,#03BH

03B6 F0 MOVX @DPTR,A

03B7 117B ACALL ESPE

03B9 740C MOV A,#00CH

03BB F0 MOVX @DPTR,A

03BC 117B ACALL ESPE

03BE 7401 MOV A,#001H

03C0 F0 MOVX @DPTR,A

03C1 7B06 MOV R3,#006H

03C3 117B CLE2: ACALL ESPE

03C5 DBFC DJNZ R3,CLE2

03C7 7406 MOV A,#006H

```
03C9 F0          MOVX @DPTR,A
03CA 117B        ACALL ESPE
03CC 22          RET
```

.....

; RUTINA DE INTERRUPCION DE ATENCIÓN EXTERNA 0

; A través de esta rutina se maneja los botones del dispositivo apuntador

;

;

.....

```
03CD 117B        RIE0: ACALL ESPE
03CF 209141      JB P1.1, MOUSB2
03D2 758968      mov tmod,#066h
03D5 758C38      mov th0,#038h
03D8 758D38      mov th1,#038h
03DB 758A38      mov ll0,#038h
03DE 758B38      mov ll1,#038h
03E1 758855      mov tcon,#055h
03E4 750300      mov 003h,#000h
03E7 750000      MOV 000H,#000H
03EA 750400      MOV 004H,#000H
03ED 750700      mov 007h,#000h
03F0 C011        PUSH 011H
03F2 C012        PUSH 012H
03F4 C0E0        PUSH 0E0H
03F8 C01F        PUSH 01FH
03F8 C0D0        PUSH PSW
03FA 7511CF      MOV 011H,#0CFH
03FD 7512F6      MOV 012H,#0F6H
0400 D2D5        SETB PSW.5
0402 91AC        ACALL DDRV
0404 D0D0        POP PSW
0406 D01F        POP 01FH
0408 D0E0        POP 0E0H
040A D012        POP 012H
040C D011        POP 011H
040E C2A8        CLR IE.0
```

```

0410 0204AB      L JMP FINT1
0413 2090B7      MOUSB2: JB P1.0,RIE0
0416 758805      MOV TCON,#005H
0419 C083        PUSH DPH
041B C082        PUSH DPL
041D C0E0        PUSH 0E0H
041F C0D0        PUSH PSW
0421 C011        PUSH 011H
0423 C012        PUSH 012H
0425 C01F        PUSH 01FH
0427 7400        MOV A,#000H
0429 900000      MOV DPTR,#0000H
042C F0          MOVX @DPTR,A
042D C3          CLR C
042E E51D        MOV A,01DH      ; Determina dirección del
archivo
0430 3401        ADDC A,#001H
0432 5002        JNC NOSIG
0434 051E        INC 01EH
0436 F51D        NOSIG: MOV 01DH,A
0438 C3          CLR C
0439 E51B        MOV A,01BH
043B 3401        ADDC A,#001H
043D 5002        JNC NOSI2
043F 051C        INC 01CH
0441 F51B        NOSI2: MOV 01BH,A
0443 E51A        MOV A,01AH      ; Lee la cantidad de archivos
0445 900001      MOV DPTR,#0001H
0448 F0          MOVX @DPTR,A   ; Almacena el número de
archivos
0449 23          RL A
044A F582        MOV DPL,A      ; Almacena el número de datos por
archivo
044C E51D        MOV A,01DH
044E F0          MOVX @DPTR,A
044F A3          INC DPTR

```

```

0450 E51E      MOV A,01EH
0452 4528      ORL A,026H
0454 F0        MOVX @DPTR,A
0455 C3        CLR C          ; Dirección de datos
0458 E51B      MOV A,01BH
0458 33        RLC A
0459 F582      MOV DPL,A
045B E51C      MOV A,01CH
045D 33        RLC A
045E F583      MOV DPH,A
0460 C3        CLR C
0461 E582      MOV A,DPL
0463 34BE      ADDC A,#0BEH
0465 F582      MOV DPL,A
0467 5002      JNC WAIT
0469 0583      INC DPH
046B 117B      WAIT: ACALL ESPE
046D 3097FB     JNB P1.7,WAIT
0470 B11E      ACALL CUENT
0472 E504      MOV A,004H
0474 F0        MOVX @DPTR,A
0475 E507      MOV A,007H
0477 A3        INC DPTR
0478 F0        MOVX @DPTR,A
0479 117B      WAIT2: ACALL ESPE
047B 3097FB     JNB P1.7,WAIT2
047E E526      MOV A,026H
0480 B4600E     CJNE A,#060H,EXIT
0483 303804     JNB 038H,HEX
0486 7405      MOV A,#005H
0488 8002      SJMP EXT
048A 7406      HEX: MOV A,#006H
048C 120918     EXT: LCALL LTR ;+++++++LETRERO ANCHO O LARGO+++
048F B238      CPL 038H
0491 7511CF     EXIT: MOV 011H,#0CFH
0494 7512E8     MOV 012H,#0E8H

```

```
0497 D2D5      SETB PSW.5
0499 91AC      ACALL DDRV
049B D01F      POP 01FH
049D D012      POP 012H
049F D011      POP 011H
04A1 D0D0      POP PSW
04A3 D0E0      POP 0E0H
04A5 D082      POP DPL
04A7 D083      POP DPH
04A9 C2A8      CLR IE.0
04AB 32        FINT1: RETI
```

```
.....
; MANEJADOR DE LA PANTALLA
```

```
; La rutina se encarga de desplegar un caracter en una posición específica
; de la pantalla sin modificar el resto de su contenido
```

```
.....
04AC C083      DDRV: PUSH DPH
04AE C082      PUSH DPL
04B0 C0D0      PUSH PSW
04B2 E525      MOV A,#025H
04B4 B40807    CJNE A,#008H,TRSM
04B7 D2D3      SETB PSW.3
04B9 C2D4      CLR PSW.4
04BB 0204C2    LJMP CDRV
04BE D2D4      TRSM: SETB PSW.4
04C0 C2D3      CLR PSW.3
04C2 902000    CDRV: MOV DPTR,#2000H
04C5 E9        MOV A,R1
04C6 F0        MOVX @DPTR,A
04C7 117B      ACALL ESPE
04C9 30D506    JNB PSW.5,REG
04CC 0582      INC DPL
04CE EA        MOV A,R2
04CF F0        MOVX @DPTR,A
04D0 117B      ACALL ESPE
```

```

04D2 D0D0      REG:  POP PSW
04D4 C2D5      CLR PSW.5
04D6 D082      POP DPL
04D8 D083      POP DPH
04DA 22        RET

```

```

;.....
; DESPLIEGUE EN HEXADECIMAL DE UN VALOR

```

```

; Se encarga de hacer la conversi3n del valor de un byte a sus caracteres
; en ASCII para su despliegue en la pantalla
;
;.....

```

```

04DB C0D0      DHEX: PUSH PSW
04DD E525      MOV A,025H
04DF B40807    CJNE A,#008H,TRSM2
04E2 D2D3      SETB PSW.3
04E4 C2D4      CLR PSW.4
04E6 0204ED    LJMP CDHE
04E9 D2D4      TRSM2 SETB PSW.4
04EB C2D3      CLR PSW.3
04ED EB        CDHE: MOV A,R3
04EE 540F      ANL A,#00FH
04F0 2430      ADD A,#030H
04F2 B43903    CJNE A,#039H,MAS
04F5 0204FE    LJMP DESP
04F8 4004      MAS:  JC  DESP
04FA 9409      SUBB A,#009H
04FC 2410      ADD A,#010H
04FE D2D5      DESP: SETB PSW.5
0500 FA        MOV R2,A
0501 91AC      ACALL DDRV
0503 EB        MOV A,R3
0504 C4        SWAP A
0505 540F      ANL A,#00FH
0507 2430      ADD A,#030H
0509 B43903    CJNE A,#039H,MAS2
050C 020515    LJMP DESP1

```

```
050F 4004    MAS2: JC DESP1
0511 9409          SUBB A,#009H
0513 2410          ADD A,#010H
0515 19          DESP1: DEC R1
0516 D2D5          SETB PSW.5
0518 FA          MOV R2,A
0519 91AC          ACALL DDRV
051B D0D0          POP PSW
051D 22          RET
```

.....
; RUTINA DE CUENTA DE DISTANCIA

; Esta rutina se encarga de hacer la conversión entre la cantidad de pulsos
; que ha enviado el dispositivo apuntador y la distancia recorrida
;

.....
051E C082 CUENT: PUSH DPL
0520 C083 PUSH DPH
0522 C0D0 PUSH PSW
0524 C3 CLR C
0525 C0E0 PUSH 0E0H
0527 C020 PUSH 020H
0529 752000 MOV 020H,#000H
052C C2D3 CLR PSW.3
052E C2D4 CLR PSW.4
0530 E501 MOV A,001H
0532 9502 SUBB A,002H
0534 F504 MOV 004H,A
0536 5002 JNC NNPL
0538 D207 SETB 7
053A E503 NNPL: MOV A,003H
053C 9500 SUBB A,000H
053E 4016 JC NTOT
0540 F507 MOV 007,A
0542 7511C6 MOV 011H,#0C6H
0545 7512A0 MOV 012H,#0A0H
0548 D2D5 SETB PSW.5

```
054A 91AC      ACALL DDRV
054C 30072C    JNB 7,ALFN
054F E504      MOV A,004H
0551 9438      SUBB A,#038H
0553 020579    LJMP ALFIN
0556 F4        NTOT: CPL A
0557 F507      MOV 007H,A
0559 7511C6    MOV 011H,#0C8H
055C 7512B0    MOV 012H,#0B0H
055F D2D5      SETB PSW,5
0561 91AC      ACALL DDRV
0563 300707    JNB 7,POST
0566 E504      MOV A,004H
0568 F4        CPL A
0569 04        INC A
056A 020579    LJMP ALFIN
056D BC0004    POST: CJNE R4,#000H,NOCER
0570 0F        INC R7
0571 02057B    LJMP ALFN
0574 E504      NOCER: MOV A,004H
0576 F4        CPL A
0577 9437      SUBB A,#037H
0579 F504      ALFN: MOV 004H,A
057B D020      ALFN: POP 020H
057D D0E0      POP 0E0H
057F D0D0      POP PSW
0581 D083      POP DPH
0583 D082      POP DPL
0585 22        RET
```

```
.....
; SUBROUTINA DE COMPARACION
```

```
; Esta rutina realiza la comparación de los datos y nos indica una bandera
; su estado
```

```
.....
0586 850D05    COMP: CJNE A,00DH,NOEQ
```

```
0589 750D03      MOV 00DH,#003H
058C 800A        SJMP REGR
058E 4005        NOEQ: JC MINC
0590 750D03      MOV 00DH,#003H
0593 8003        SJMP REGR
0595 750D00      MINC: MOV 00DH,#000H
0598 22          REGR: RET
```

.....
; RUTINA DE CONVERSION HEX-BCD

; Tiene el fin de convertir los valores almacenados en binario a valores en
; con representación decimal

.....
0599 C0D0 HEXBCD: PUSH PSW
059B D2D3 SETB PSW.3
059D C2D4 CLR PSW.4
059F 118A ACALL PERRO
05A1 7900 MOV R1,#000H
05A3 7A00 MOV R2,#000H
05A5 FB LOOPH: MOV R3,A
05A6 E9 MOV A,R1
05A7 7D05 MOV R5,#05H
05A9 B186 ACALL COMP
05AB 2D ADD A,R5
05AC F9 MOV R1,A
05AD EA MOV A,R2
05AE 7D05 MOV R5,#05H
05B0 B186 ACALL COMP
05B2 2D ADD A,R5
05B3 FA MOV R2,A
05B4 E9 MOV A,R1
05B5 23 RL A
05B6 F9 MOV R1,A
05B7 EA MOV A,R2
05B8 23 RL A
05B9 FA MOV R2,A

```
05BA 5410      ANL A,#10H
05BC 6005      JZ     NCR1
05BE EA        MOV A,R2
05BF 9410      SUBB A,#10H
05C1 09        INC R1
05C2 FA        MOV R2,A
05C3 EB        NCR1: MOV A,R3
05C4 33        RLC A
05C5 5001      JNC NCR2
05C7 0A        INC R2
05C8 DCDB      NCR2: DJNZ R4,LOOPH 05CA EA MOV A,R2
05CB 2430      ADD A,#030H
05CD F512      MOV 012H,A
05CF D2D5      SETB PSW.5
05D1 850F11    MOV 011H,00FH
05D4 91AC      ACALL DDRV
05D6 E9        MOV A,R1
05D7 2430      ADD A,#030H
05D9 F512      MOV 012H,A
05DB D2D5      SETB PSW.5
05DD 150F      DEC 00FH
05DF 850F11    MOV 011H,00FH
05E2 91AC      ACALL DDRV
05E4 D0D0      POP PSW
05E6 22        RET
05E7 7511C2    DESCU: MOV 011H,#0C2H
05EA 85A1A3    MOV 013H,01AH
05ED 91DB      ACALL DHEX
05EF 7511B4    MOV 011H,#084H
05F2 851B13    MOV 013H,01BH
05F5 91DB      ACALL DHEX
05F7 7511B2    MOV 011H,#082H
05FA 851C13    MOV 013H,01CH
05FD 91DB      ACALL DHEX
05FF C3        CLR C
0600 E58A      MOV A,TL0
```

```
0602 9438      SUBB A,#038H
0604 F9        MOV R1,A
0605 C3        CLR C
0606 E58B      MOV A,TL1
0608 9438      SUBB A,#038H
060A FA        MOV R2,A
060B B11E      ACALL CUENT
060D 750C08    MOV 00CH,#008H
0610 750FC8    MOV 00FH,#0C8H
0613 EF        MOV A,R7
0614 B199      ACALL HEXBCD
0616 750C07    MOV 00CH,#007H
0619 750FCB    MOV 00FH,#0CBH
061C EC        MOV A,R4
061D B199      ACALL HEXBCD
061F 8C20      MOV 020H,R4
0621 751230    MOV 012H,#030H
0624 300003    JNB 0,NPUC
0627 751235    MOV 012H,#035H
062A 7511CC    NPUC: MOV 011H,#0CCH
062D D2D5      SETB PSW.5
062F 91AC      ACALL DDRV
0631 118A      ACALL PERRO
0633 7511C3    MOV 011H,#0C3H
0636 85A812    MOV 012H,IE
0639 D2D5      SETB PSW.5
063B 91AC      ACALL DDRV
063D 20A80F    JB IE.0,B0
0640 20B204    JB P3.2,SOLTO
0643 7E05      MOV R8,#005H
0645 8008      SJMP B0
0647 BE0004    SOLTO: CJNE R6,#000H,ESPET3
064A D2A8      SETB IE.0      ; Habilita toda Int.
064C 8001      SJMP B0
064E 1E        ESPET3: DEC R6
064F 22        B0:   RET
```

.....
; RUTINA DE TOMA DE DATOS

; Esta es la rutina que inicia la operación del dispositivo apuntador para
; la adquisición de datos.

.....
0650 752504 DATOS: MOV 025H,#004H
0653 758A38 MOV TL0,#038H
0656 758B38 MOV TL1,#038H
0659 758805 MOV TCON,#005H
065C 752000 MOV 020H,#000H
065F C2D4 CLR PSW.4
0661 C2D3 CLR PSW.3
0663 7800 MOV R0,#000H
0665 7C00 MOV R4,#000H
0667 7D00 MOV R5,#000H
0669 7F00 MOV R7,#000H
066B 75A88F MOV IE,#08FH
066E 75B805 MOV IP,#005H
0671 751B00 MOV 01BH,#000H
0674 751C00 MOV 01CH,#000H
0677 751D00 MOV 01DH,#000H
067A 751E00 MOV 01EH,#000H
067D 752600 MOV 026H,#000H
0680 7402 MOV A,#002H
0682 120956 LCALL IMPSP ;+++++++ ESCOGE EL TIPO DE
DATOS++++
0685 B40107 CJNE A,#001H,SIG
0688 752620 MOV 026H,#020H
068B 7405 MOV A,#005H
068D 8015 SJMP CTR
068F B40207 SIG: CJNE A,#002H,SIG2
0692 752640 MOV 026H,#040H
0695 7406 MOV A,#006H
0697 800B SJMP CTR
0699 752700 SIG2: MOV 027H,#000H

```

069C 7405      MOV A,#005H
069E 752680    MOV 028H,#060H
06A1 752700    MOV 027H,#000H
06A4 120918    CTR:  LCALL LTR ;+++++IMPRIME LETRERO ANCHO O
LARGO+++
06A7 752400    MOV 024H,#000H
06AA 900000    MOV DPTR,#0000H      ; Revisión de la memoria
06AD E0        MOVX A,@DPTR
06AE 7511CE    MOV 011H,#0CEH
06B1 F512      MOV 012H,A
06B3 D2D5      SETB PSW.5
06B5 91AC      ACALL DDRV
06B7 E512      MOV A,012H
06B9 6005      JZ NZ
06BB 751A01    MOV 01AH,#001H
06BE 800A      SJMP LOOP
06C0 A3        NZ:   INC DPTR      ; Determina la dirección del siguiente
dato
06C1 E0        MOVX A,@DPTR
06C2 C3        CLR C
06C3 F508      MOV 008H,A
06C5 04        INC A
06C6 F51A      MOV 01AH,A
06C8 7180      ACALL ADDAT
06CA B1E7      LOOP: ACALL DESCU
06CC E524      MOV A,024H
06CE B40003    CJNE A,#000H,REGT
06D1 0206CA    LJMP LOOP
06D4 22        REGT: RET
.....
; SUBROUTINA DE RECEPCION
; Esta rutina se encarga de transmitir un buffer de datos.
; En caso de que detecte algún error retorna una bandera
;
.....
06D5 AB05      RX:   MOV R3,R5

```

```
06D7 7CFF          MOV R4,#0FFH
06D9 7818          MOV R0,#018H
06DB 118A          NCOM: ACALL PERRO
06DD 3098FB        JNB SCON.0,NCOM
06E0 C298          CLR SCON.0
06E2 AC99          MOV R4,SBUF
06E4 E6           MOV A,@R0
06E5 B50405        CJNE A,R4,RETU
06E8 08           INC R0
06E9 DBF0          DJNZ R3,NCOM
06EB D2D5          SETB PSW.5
06ED 22           RETU: RET
```

```
.....
; SUBROUTINA DE TRANSMISION
```

```
; Se encarga de la transmisión de un buffer de datos
```

```
.....
06EE A805          TX:   MOV R3,R5
06F0 7810          MOV R0,#010H
06F2 118A          TRANS: ACALL PERRO
06F4 852817        MOV 017H,028H
06F7 D51702        ORTR: DJNZ 017H,SINC
06FA 8005          SJMP NSP
06FC 12070E        SINC: LCALL SPE
06FF 80F6          SJMP ORTR
0701 E6           NSP:  MOV A,@R0
0702 F599          MOV SBUF,A
0704 00           TCOM: NOP
0705 3099FC        JNB SCON.1,TCOM
0708 08           INC R0
0709 C299          CLR SCON.1
070B DBE5          DJNZ R3,TRANS
070D 22           RET
```

```
.....
; RUTINA DE ESPERA
```

; Esta rutina detienen el procesador durante menos tiempo

```

070E 12008A  SPE:  LCALL PERRO
0711 751FFF  MOV 01FH,#0FFH
0714 D51FFD  KSP:  DJNZ 01FH,KSP
0717 12008A          LCALL PERRO
071A 22          RET

```

; PROCEDIMIENTO DE COMUNICACION

; Se encarga de manejar todo el protocolo de comunicaci3n para la transmisi3n
; de los datos

```

071B 118A      SCOM:  ACALL PERRO
071D C2D4          CLR PSW.4           ; Define banco de registros 0
071F C2D3          CLR PSW.3
0721 752508      MOV 025H,#008H
0724 759850      MOV SCOM,#050H
0727 75A884      MOV IE,#084H
072A 758920      MOV TMOD,#020H
072D 758840      MOV TCON,#040H
0730 750C00      MOV 00CH,#000H
0733 750D00      MOV 00DH,#000H
0736 750E00      MOV 00EH,#000H
0739 752890      MOV 028H,#090H
073C 7410          MOV A,#010H
073E 120956      LCALL  IMPSP      ;+++++++VELOCIDAD  DE
TRANSMISION++++
0741 14          DEC A
0742 23          RL A
0743 23          RL A
0744 23          RL A
0745 900749      MOV DPTR,#(CASE6)H
0748 73          JMP @A+DPTR

```

```

0749 758DE8 CASE6: MOV TH1,#0E8H
074C 758BE8      MOV TL1,#0E8H
074F 800E      SJMP YCOM
0751 758DF4      MOV TH1,#0F4H
0754 758BF4      MOV TL1,#0F4H
0757 8006      SJMP YCOM
0759 758DFA      MOV TH1,#0FAH
075C 758BFA      MOV TL1,#0FAH
075F 7407      YCOM: MOV A,#007H
0761 120918      LCALL LTR ;+++++++LETRERO DE
COMUNICACION+++++
0764 118A      MAL: ACALL PERRO
0766 75A800      MOV IE,#000H
0769 7D04      MOV R5,#004H
076B 751800      MOV 018H,#000H ; Espera la recepción de
reconocimiento
076E 751900      MOV 019H,#000H
0771 751A50      MOV 01AH,#050H
0774 751B43      MOV 01BH,#043H
0777 1206D5      LCALL RX
077A 30D5E7      JNB PSW.5,MAL
077D C2D5      CLR PSW.5
077F 7D04      MOV R5,#004H
0781 751000      MOV 010H,#000H ; Transmite mensaje de
reconocimiento y aceptación
0784 751100      MOV 011H,#000H
0787 751233      MOV 012H,#033H
078A 751331      MOV 013H,#031H
078D 1208EE      LCALL TX
0790 7D03      MOV R5,#003H
0792 751800      MOV 018H,#000H ; Espera respuesta de
confirmación
0795 751900      MOV 019H,#000H
0798 751A02      MOV 01AH,#002H
079B 1206D5      LCALL RX
079E 30D5C3      JNB PSW.5,MAL

```

```

07A1 C2D5          CLR PSW.5
07A3 12008A      ESRET: LCALL PERRO
07A6 3098FA      JNB SCON.0,ESRET
07A9 C298        CLR SCON.0
07AB E599        MOV A,SBUF
07AD B4FF03      CJNE A,#0FFH,CTRM
07B0 0208DB      LJMP OPTI
07B3 859923      CTRM: MOV 028H,SBUF
07B6 7411        MOV A,#011H
07B8 120918      LCALL          LTR          ;*****ENLACE
ESTABLECIDO*****
07BB 900000      MOV DPTR,#0000H
07BE E0          MOVX A,@DPTR
07BF B40002      CJNE A,#000H,ERME
07C2 8003        SJMP CTIN
07C4 0208AC      ERME: LJMP ERMEM
07C7 A3          CTIN: INC DPTR
07C8 E0          MOVX A,@DPTR
07C9 F508        MOV 008H,A
07CB 751000      MOV 010H,#000H ; Transmite el número de
archivos
07CE 751100      MOV 011H,#000H
07D1 F512        MOV 012H,A
07D3 7D03        MOV R5,#003H
07D5 1206EE      LCALL TX
07D8 751800      MOV 018H,#000H ; Espera confirmación
07DB 751900      MOV 019H,#000H
07DE 751A02      MOV 01AH,#002H
07E1 7D03        MOV R5,#003H
07E3 1206D5      LCALL RX
07E6 30D532      JNB PSW.5,MAL2
07E9 C2D5        CLR PSW.5
07EB A3          CTM2: INC DPTR
07EC E0          MOVX A,@DPTR
07ED F510        MOV 010H,A ;Carga el primer dato en el buffer de
transmisión

```

| | | |
|-------------------|--------------------|---|
| 07EF C3 | CLR C | |
| 07F0 350D | ADDC A,00DH | |
| 07F2 F50D | MOV 00DH,A | |
| 07F4 A3 | INC DPTR | |
| 07F5 E0 | MOVX A,@DPTR | |
| 07F6 F511 | MOV 011H,A | ;Carga el segundo dato en el buffer de |
| transmisión | | |
| 07F8 541F | ANL A,#01FH | ; Elimina el tipo de archivo y almacena |
| solo la cantidad. | | |
| 07FA 350E | ADDC A,00EH | |
| 07FC F50E | MOV 00EH,A | |
| 07FE 7D02 | MOV R5,#002H | |
| 0800 1206EE | LCALL TX | ; Transmite cantidad de datos de |
| archivos | | |
| 0803 D508E5 | DJNZ 008H,CTM2 | |
| 0806 751800 | MOV 018H,#000H | ; Espera confirmación |
| 0809 751900 | MOV 019H,#000H | |
| 080C 751A02 | MOV 01AH,#002H | |
| 080F 7D03 | MOV R5,#003H | |
| 0811 1206D5 | LCALL RX | |
| 0814 30D504 | JNB PSW.5,MAL2 | |
| 0817 C2D5 | CLR PSW.5 | |
| 0819 8003 | SJMP CNSR | |
| 081B 020764 | MAL2: LJMP MAL | |
| 081E 7D04 | CNSR: MOV R5,#004H | |
| 0820 751000 | MOV 010H,#000H | |
| 0823 751100 | MOV 011H,#000H | |
| 0826 850E12 | MOV 012H,00EH | |
| 0829 850D13 | MOV 013H,00DH | |
| 082C 1206EE | LCALL TX | ; Transmite el total de datos a |
| transmitir | | |
| 082F 7D03 | MOV R5,#003H | |
| 0831 751800 | MOV 018H,#000H | |
| 0834 751900 | MOV 019H,#000H | |
| 0837 751A02 | MOV 01AH,#002H | |
| 083A 1206D5 | LCALL RX | |

```
083D 30D569      JNB PSW.5,MAL1
0840 C2D5        CLR PSW.5
0842 7412        MOV A,#012H
0844 120918      LCALL LTR ;*****INICIA  TRANSMICION  DE
DATOS*****
0847 C3          CLR C
0848 E50D        MOV A,#00DH
084A 33          RLC A
084B F582        MOV DPL,A
084D E50E        MOV A,#00EH
084F 33          RLC A
0850 F583        MOV DPH,A
0852 C3          CLR C
0853 E582        MOV A,DPL
0855 34BF        ADDC A,#0BFH
0857 5002        JNC NSIH
0859 0583        INC DPH
085B F582      NSIH: MOV DPL,A
085D 858206      MOV 008H,DPL
0860 858307      MOV 007H,DPH
0863 9000BF      MOV DPTR,#00BFH
0866 A3          SIGDA: INC DPTR
0867 E0          SIGDAV: MOVX A,@DPTR
0868 F511        MOV 011H,A
086A A3          INC DPTR
086B E0          MOVX A,@DPTR
086C F510        MOV 010H,A
086E 7D02        MOV R5,#002H
0870 1206EE      LCALL TX
0873 309803      JNB SCON.0,CON
0876 020764      LJMP MAL
0879 E583      CON:  MOV A,DPH
087B B507E8      CJNE A,007H,SIGDA
087E E582        MOV A,DPL
0880 B506E3      CJNE A,006H,SIGDA
0883 7D03        MOV R5,#003H
```

```

0885 751000      MOV 010H,#000H
0888 751100      MOV 011H,#000H
088B 751203      MOV 012H,#003H
088E 1208EE      LCALL TX
0891 7D04        MOV R5,#004H
0893 751800      MOV 018H,#000H
0896 751900      MOV 019H,#000H
0899 850D1A      MOV 01AH,00DH
089C 850C1B      MOV 01BH,00CH
089F 1206D5      LCALL RX
08A2 30D504      JNB PSW.5,MAL1
08A5 12008A      FIN:  LCALL PERRO
08A8 22          RET
08A9 020764      MAL1: LJMP MAL
08AC 7408        ERMEM: MOV A,#008H
08AE 120918      LCALL  LTR      ;+++++ERROR      EN
MEMORIA+++++
08B1 751000      MOV 010H,#000H
08B4 7511FF      MOV 011H,#0FFH
08B7 7512FF      MOV 012H,#0FFH
08BA 7D03        MOV R5,#003H
08BC 1208EE      LCALL TX
08BF 751800      MOV 018H,#000H
08C2 751900      MOV 019H,#000H
08C5 751A02      MOV 01AH,#002H
08C8 7D03        MOV R5,#003H ; Elige transmitir datos sin formato
08CA 1206D5      LCALL RX
08CD 30D5D5      JNB PSW.5,FIN
08D0 750EFF      SIGCM: MOV 00EH,#0FFH
08D3 750F1F      MOV 00FH,#01FH
08D6 900000      MOV DPTR,#0000H
08D9 808C        SJMP SIGDAV
08DB 752601      OPTI: MOV 026H,#001H
08DE 7414        MOV A,#014H
08E0 120918      LCALL  LTR      ;+++++OPTIMIZANDO
COMUNICACION++++

```

```

08E3 751130   TRDN:  MOV 011H,#030H
08E6 7512DF           MOV 012H,#0DFH
08E9 7D02           MOV R5,#002H
08EB 1206EE           LCALL TX
08EE 751800           MOV 018H,#000H
08F1 751900           MOV 019H,#000H
08F4 751A02           MOV 01AH,#002H
08F7 7D03           MOV R5,#003H;Verifica si la informacion llego bien
08F8 1206D5           LCALL RX
08FC 30D50B           JNB PSW.5,FINS
08FF 852811           MOV 011H,028H
0902 7D01           MOV R5,#001H
0904 1206EE           LCALL TX
0907 0208A5           LJMP FIN
090A 0526           FINS:  INC 026H
090C E526           MOV A,026H
090E B4FFD2           CJNE A,#0FFH,TRDN
0911 7415           MOV A,#015H
0913 3156           ACALL IMPSP
0915 0208A5           LJMP FIN

```

.....

; SUBROUTINA DE DÉSPLIEGUE DE LETREROS

; Esta rutina se encarga la obtención de la dirección del letrero a partir de
; su número

.....

```

0918 C082   LTR:  PUSH DPL
091A C083           PUSH DPH
091C 23           RL A           ; Calculo de dirección del letrero
091D 24A0           ADD A,#0A0H
091F F5F0           MOV B,A
0921 540F           ANL A,#00FH
0923 C4           SWAP A
0924 F582           MOV DPL,A
0926 E5F0           MOV A,B
0928 54F0           ANL A,#0F0H

```

```

092A 2404      ADD A,#004H
092C C4        SWAP A
092D F583      MOV DPH,A
092F 7480      MOV A,#080H
0931 3162      ACALL TXLTR
0933 74C0      MOV A,#0C0H
0935 3162      ACALL TXLTR
0937 D083      POP DPH
0939 D082      POP DPL
093B 22        RET

```

```

;.....
; RUTINA DE DESPLIEGUE LARGO

```

```

; En caso de tratarse de un desplegado de opciones se realiza la impresion
; del letrero y se detiene toda la operación del sistema mientras se
; obtiene un resultado
;.....

```

```

093C 3118      LDSP: ACALL LTR
093E 751F0F     MOV 01FH,#00FH
0941 C01F      LDS2: PUSH 01FH
0943 751FFF     MOV 01FH,#0FFH
0946 C01F      LDS1: PUSH 01FH
0948 12007B     LCALL ESPE
094B D01F      POP 01FH
094D D51FF8     DJNZ 01FH, LDS1
0950 D01F      POP 01FH
0952 D51FEC     DJNZ 01FH, LDS2
0955 22        RET
0956 3118      IMPSP: ACALL LTR
0958 C295      IMPS: CLR P1.5          ; Detener el "watch dog"
095A D295      SETB P1.5             ; Reactivar el "watch dog"
095C 438701     ORL PCON,#001H; conjelar el 8051 mientras llega la
interrupcion
095F E524      MOV A,024H
0961 22        RET

```

```

;.....
; SUBROUTINA DE IMPRESION DE LETRERO

```

; La rutina se encarga de leer cada carácter de la memoria y desplegarlo
; en la pantalla

.....

| | | |
|-------------|--------------------|-----------------------|
| 0962 C082 | TXLTR: PUSH DPL | |
| 0964 C083 | PUSH DPH | |
| 0966 902000 | MOV DPTR,#2000H | |
| 0969 F0 | MOVX @DPTR,A | |
| 096A D083 | POP DPH | |
| 096C D082 | POP DPL | |
| 096E 12007B | LCALL ESPE | |
| 0971 75F010 | MOV B,#010H | ; Envío de caracteres |
| 0974 E0 | NCAR: MOVX A,@DPTR | |
| 0975 C082 | PUSH DPL | |
| 0977 C083 | PUSH DPH | |
| 0979 902001 | MOV DPTR,#2001H | |
| 097C F0 | MOVX @DPTR,A | |
| 097D 12007B | LCALL ESPE | |
| 0980 D083 | POP DPH | |
| 0982 D082 | POP DPL | |
| 0984 A3 | INC DPTR | |
| 0985 D5F0EC | DJNZ B,NCAR | |
| 0988 22 | RET | |
| 0000 | END | |