

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería en Electrónica**



**Instituto Costarricense de Electricidad  
ICE**

**Diseño e implementación de un patrón nacional de tiempo a partir de un  
oscilador de cesio de 10 MHz**

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en  
Electrónica con el Grado Académico de Licenciatura**

**Adriana Alvarado Salazar**

**Cartago, II Semestre 2002**

## *DEDICATORIA*

*A Dios, por brindarme sabiduría, fortaleza y entendimiento para sobrellevar todas las situaciones de mi vida. A mi familia por su apoyo incondicional, amor y comprensión. A Juan Francisco y nuestra nueva familia, mi nueva motivación de constante crecimiento y lucha.*

## **Agradecimiento**

Agradezco al Dr. Harold Sánchez, Director del Laboratorio Metrológico de Variables Eléctricas, su ayuda y comprensión durante el desarrollo del proyecto de graduación. Asimismo, agradezco al Ing. Miguel Hernández, su disposición al diálogo y organización.

Finalmente, agradezco al Ing. Carlos Badilla la oportunidad dada para hacer realidad, el sueño de estudiar Ingeniería Electrónica en el ITCR.

## INDICE

|   |           |
|---|-----------|
| <b><u>Capítulo 1: Introducción</u></b> .....  | <b>11</b> |
| <b><u>1.1 Descripción de la empresa</u></b> .....   | <b>11</b> |
| <b><u>1.1.1 Descripción general</u></b> .....   | <b>11</b> |
| <b><u>1.1.2 Descripción del departamento donde se realizó el proyecto de graduación</u></b> ..... | <b>12</b> |
| <b><u>1.2 Definición del problema y su importancia</u></b> .....                                  | <b>14</b> |
| <b><u>1.3 Objetivos</u></b> .....   | <b>17</b> |
| <b><u>1.3.1 Objetivo general</u></b> .....  | <b>17</b> |
| <b><u>1.3.2 Objetivos específicos</u></b> .....   | <b>17</b> |
| <b><u>Capítulo 2: Antecedentes</u></b> .....  | <b>20</b> |
| <b><u>2.1 Estudio del problema a resolver</u></b> .....   | <b>20</b> |
| <b><u>2.2 Requerimientos de la empresa</u></b> .....  | <b>22</b> |
| <b><u>2.3 Solución Propuesta</u></b> .....  | <b>24</b> |
| <b><u>Capítulo 3: Procedimiento metodológico</u></b> .....  | <b>27</b> |
| <b><u>3.1 Procedimiento metodológico</u></b> .....  | <b>27</b> |
| <b><u>Capítulo 4: Descripción del hardware utilizado</u></b> .....                                | <b>38</b> |
| <b><u>4.1 Descripción del hardware utilizado</u></b> .....  | <b>38</b> |
| <b><u>4.1.1 Equipo e Instrumentación</u></b> .....  | <b>38</b> |
| <b><u>Capítulo 5: Descripción del software utilizado</u></b> .....                                | <b>56</b> |
| <b><u>5.1 Descripción del software del sistema</u></b> .....                                      | <b>56</b> |

|   |     |
|---|-----|
| <a href="#"><u>5.2 Descripción del software de bajo nivel para generar la interrupción del bus ISA (lectura o transmisión de datos desde el bus hacia la aplicación de la PC)</u></a> ..... | 60  |
| <a href="#"><u>5.3 Descripción del software utilizado para desarrollar la interfaz gráfica del reloj de cesio</u></a> .....   | 61  |
| <a href="#"><u>Capítulo 6: Resultados Experimentales</u></a> .....  | 72  |
| <a href="#"><u>6.1 Oscilogramas del sistema</u></a> .....   | 72  |
| <a href="#"><u>6.2 Pruebas de exactitud</u></a> .....   | 74  |
| <a href="#"><u>Capítulo 7: Análisis de resultados</u></a> .....   | 76  |
| <a href="#"><u>7.1 Explicación del diseño</u></a> .....   | 76  |
| <a href="#"><u>7.2 Alcances y limitaciones</u></a> .....  | 81  |
| <a href="#"><u>Conclusiones y Recomendaciones</u></a> .....   | 84  |
| <a href="#"><u>Bibliografía</u></a> .....   | 88  |
| <a href="#"><u>Apéndices</u></a> .....  | 90  |
| <a href="#"><u>Apéndice A.1 Glosario de términos y Lista de Abreviaturas</u></a> .....  | 91  |
| <a href="#"><u>Apéndice A.2 Circuito preliminar para la tarjeta de adquisición de datos</u></a> .....   | 93  |
| <a href="#"><u>Apéndice A.3 Manual del usuario</u></a> .....  | 95  |
| <a href="#"><u>Anexos</u></a> .....   | 107 |
| <a href="#"><u>Anexo 1 Aspectos más relevantes del PIC 16F877</u></a> .....   | 108 |
| <a href="#"><u>Anexo 2 Aspectos más relevantes del LM 7805C</u></a> .....   | 109 |

## INDICE DE FIGURAS

|  |    |
|--|----|
| <a href="#"><u>Figura 1</u></a> <a href="#"><u>Requerimientos del sistema de instalación</u></a> .....   | 15 |
| <a href="#"><u>Figura 2</u></a> <a href="#"><u>Tareas realizadas para obtener la solución</u></a> .....  | 28 |
| <a href="#"><u>Figura 4.1</u></a> <a href="#"><u>Precisión ofrecida por defecto por la PC, en la toma de muestras consecutivas</u></a> .....                               | 41 |
| <a href="#"><u>Figura 4.2</u></a> <a href="#"><u>Configuración interna del bus ISA</u></a> .....   | 42 |
| <a href="#"><u>Figura 4.3</u></a> <a href="#"><u>Arquitectura Harvard para el microcontrolador 16F877</u></a> .....  | 44 |
| <a href="#"><u>Figura 4.4</u></a> <a href="#"><u>Segmentación en el microcontrolador 16F877</u></a> .....  | 44 |
| <a href="#"><u>Figura 4.5</u></a> <a href="#"><u>Fotografía de módulo de adquisición y transmisión de datos (protoboard). Vista superior</u></a>                           | 50 |
| <a href="#"><u>Figura 4.6</u></a> <a href="#"><u>Fotografía de módulo de adquisición y transmisión de datos (protoboard). Vista lateral (conectores BNC)</u></a>           | 50 |
| <a href="#"><u>Figura 4.7</u></a> <a href="#"><u>Diagrama de bloques para el módulo de adquisición y transmisión de datos (protoboard)</u></a>                             | 51 |
| <a href="#"><u>Figura 5.1</u></a> <a href="#"><u>Diagrama de desarrollo del MPLAB</u></a>  | 57 |
| <a href="#"><u>Figura 5.2</u></a> <a href="#"><u>Distribución y asignación de los bits del Registro de opciones ..</u></a>   | 58 |
| <a href="#"><u>Figura 5.3</u></a> <a href="#"><u>Diagrama general para la programación del PIC 16F877</u></a>  | 59 |
| <a href="#"><u>Figura 5.4</u></a> <a href="#"><u>Entorno típico de Delphi 5i en la realización y ejecución de programas</u></a>  | 62 |
| <a href="#"><u>Figura 5.5</u></a> <a href="#"><u>Interfaz gráfica del reloj patrón a partir del oscilador de cesio de 10Mhz</u></a>  | 65 |
| <a href="#"><u>Figura 5.6</u></a> <a href="#"><u>Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Ventana de inicio)</u></a>                       | 67 |
| <a href="#"><u>Figura 5.7</u></a> <a href="#"><u>Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Después de presionar la opción “Cargar”)</u></a> | 67 |

|   |     |
|---|-----|
| <a href="#"><u>Figura 5.8</u></a> Opción configuración de tiempo de muestreo del menú de la base de datos   | 69  |
| <a href="#"><u>Figura 5.9</u></a> Inspector de recursos de hardware del reloj nacional a partir del oscilador de cesio de 10 MHz                      | 70  |
| <a href="#"><u>Figura 5.10</u></a> Interfaz para la clave de acceso de salida del reloj nacional a partir del oscilador de cesio de 10 MHz            | 71  |
| <a href="#"><u>Figura 6.1</u></a> Oscilograma a partir del oscilador de cesio de 10 MHz. (Sin diodos Zener)   | 72  |
| <a href="#"><u>Figura 6.2</u></a> Oscilograma a partir del oscilador de cesio de 10 MHz. (Con diodos Zener)   | 73  |
| <a href="#"><u>Figura 6.3</u></a> Oscilograma de la salida de 1 Hz  | 74  |
| <a href="#"><u>Figura A.4.1</u></a> Interfaz gráfica del reloj patrón a partir del oscilador de cesio de 10MHz  | 98  |
| <a href="#"><u>Figura A.4.2</u></a> Items de selección en el menú principal (Opciones)  | 98  |
| <a href="#"><u>Figura A.4.3</u></a> Interfaz para la clave de acceso de salida del reloj nacional a partir del oscilador de cesio de 10 MHz           | 100 |
| <a href="#"><u>Figura A.4.4</u></a> Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Ventana de inicio)                       | 100 |
| <a href="#"><u>Figura A.4.5</u></a> Inspector de recursos de hardware del reloj nacional a partir del oscilador de cesio de 10 MHz                    | 101 |
| <a href="#"><u>Figura A.4.6</u></a> Teclas de acceso rápido del reloj nacional a partir del oscilador de cesio de 10 MHz                              | 102 |
| <a href="#"><u>Figura A.4.7</u></a> Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Después de presionar la opción “Cargar”) | 103 |
| <a href="#"><u>Figura A.4.8</u></a> Opción configuración de tiempo de muestreo del menú de la base de datos   | 104 |
| <a href="#"><u>Figura A.4.9</u></a> Modo de búsqueda del bus ISA dentro del inspector de recursos de hardware   | 105 |

## INDICE DE TABLAS

|   |    |
|---|----|
| <u><a href="#">Tabla 4.1</a></u> Asignación de IRQ  | 40 |
| <u><a href="#">Tabla 4.2</a></u> Listado de las principales señales para el bus ISA                   | 43 |
| <u><a href="#">Tabla 6.1</a></u> Porcentaje de error de valores obtenidos en la salida de 1 Hz        | 75 |
| <u><a href="#">Tabla 6.2</a></u> Valores de tiempo para el almacenamiento de la base de datos         | 75 |
| <u><a href="#">Tabla A.2.1</a></u> Listado de las principales señales para la interfaz con el bus ISA | 93 |



## ***RESUMEN***

---

Los estándares de tiempo y frecuencia influyen en nuestra vida de diversas formas. A nivel mundial, el NIST provee la mayoría de los patrones de comparación, en lo que a tiempo y frecuencia respecta.

En Costa Rica, algunos de los patrones más importantes establecidos como patrones nacionales; se miden en el Laboratorio Metrológico como lo son los de tiempo, corriente, voltaje, factor de potencia y nivel de frecuencia. Sin embargo, el patrón nacional de tiempo, se mide por medio del servicio que brinda el NIST; el cual se realiza por medio de señales GPS. Así, en Costa Rica no se había desarrollado un sistema local que brindara un patrón confiable y alternativo de medición de tiempo. Este se convirtió en el objetivo de este proyecto.

El desarrollo del siguiente trabajo, representa una solución para la toma de los mediciones de tiempo y su trazabilidad con los patrones del Instituto Nacional de Estándares y Tecnologías (NIST). Como parte de la solución preliminar, se involucró la utilización de los sistemas digitales. Con los recursos que proporciona una computadora y su capacidad de interacción con dispositivos periféricos tales como el bus ISA, se obtuvo una solución satisfactoria. Dentro de las facilidades que ofrece la computadora, está la de poder manipular los dispositivos que se encargan de los procesos de temporización interna de la PC. Los dispositivos diseñados contaron con una tarjeta compatible con el bus ISA que involucró circuitos de protección. Asimismo, para implementar un sistema de alta precisión, se utilizaron los procedimientos de calibración de cronómetros del Laboratorio de Metrología para corregir el error o desviación introducido por la computadora en la interfaz gráfica diseñada.

Palabras claves: Laboratorio de Metrología; Temporización; Circuitos de protección; Desviación; NIST; Trazabilidad; Patrón de tiempo.

## **SUMMARY**

---

Time and frequency standards have different effects in our daily routine. Around the world NIST (National Institute of Standards and Technology) provides almost standards of time and frequency.

Costa Rica has many national standards and all of them are measured inside Laboratory of Metrology . These standards are time, frequency, current, voltage, etc. However national standard of time are measure with a NIST's system and GPS signals. Therefore, Costa Rica has never been a local system of time, and that was the prompt of this project.

Inside Costa Rica, time and frequency standards had been maintained by NIST for a long time. Nowadays, Costa Rica needs another system of comparison besides the provided it by NIST. This is the goal of this report. This work presents a solution for the measurements of time and their traceability to the National Time and Frequency Standards of Instituto Costarricense de Electricidad.

It was decided to use PC based systems due to the resources available in these. These resources include manipulation of timing devices and communications with external circuitry via ISA bus. This circuit also provides protection against short circuits and over voltages. The correction in precision of the PC based clock implemented was checked against time standards of the Metrology Laboratory. The programming allows this correction to be included in the calculations in order to have a minimal error in the reported results.

Key words: Metrology; Timing; NIST; Protection circuits; Deviation; Traceability.

## **CAPÍTULO 1** **INTRODUCCIÓN**

---

### **1.1 Descripción de la empresa**

#### **1.1.1 Descripción general**

El Instituto Costarricense de Electricidad (ICE) fue creado el 8 de Abril de 1949, bajo el Decreto de Ley No 449, denominado “Ley de Creación del Instituto Costarricense de Electricidad”, de la Junta Fundadora de la Segunda República. Su propósito inicial era resolver los problemas de suministro de energía eléctrica que afectaban al país, pues este servicio lo ofrecían empresas privadas sin ningún interés en desarrollarlo.

Además de ser una institución autónoma del Estado con personería jurídica propia, el Instituto Costarricense de Electricidad (ICE) se financia con fondos propios y externos, a través de la venta de servicios de energía y telecomunicaciones. La globalización de los mercados y la revolución tecnológica llevan a las empresas del Grupo ICE a redoblar esfuerzos con una clara orientación hacia el cliente, con los mejores y más innovadores productos y servicios; utilizando menos recursos y tiempo. Con más de medio siglo de existencia, el ICE ha logrado la construcción de numerosas obras hidroeléctricas, térmicas y geotérmicas; instalación de paneles solares en comunidades alejadas y la producción de energía eólica, y otros. Producto de estos logros, la generación eléctrica logró dar cobertura eléctrica al 96.8% del territorio nacional.

Al crecer las exigencias del mercado mundial en el suministro de energía eléctrica y comunicaciones, el ICE ha incursionado en otras áreas de demanda nacional. Es así como en 1963, se le asignó los servicios de telecomunicaciones de acuerdo con la Ley No 3226. En el sector de Telecomunicaciones, el ICE ha alcanzado una posición privilegiada con respecto a Latinoamérica, pues cuenta con 24.37 teléfonos convencionales por cada 100 habitantes. Costa Rica posee una

cobertura telefónica del 94% de la población, colocándolo en el primer lugar de los países centroamericanos y el segundo a nivel sudamericano. Ofrece un servicio de telefonía básica, con los precios más bajos de instalación y tarifa mensual, así como otras facilidades avanzadas de telefonía (servicios de telecomunicaciones).

En el sector de las telecomunicaciones (ICETEL), se cuenta con el Centro Funcional 6693, ubicado en San Pedro. Dentro de éste edificio, se ubica mediante convenio celebrado entre el Ministerio de Economía, Industria y Comercio (MEIC) y el Instituto Costarricense de Electricidad (ICE), el Laboratorio Metrológico de Variables Eléctricas, que será el departamento encargado de guiar y asesorar el Proyecto de Graduación.

### **1.1.2 Descripción del departamento donde se realizó el proyecto de graduación**

El Laboratorio Metrológico de Variables Eléctricas participa en el Sistema Interamericano de Metrología (SIM) y mantiene una relación preferente de referencia con el National Institute of Standards and Technologies (NIST) de los Estados Unidos de América. Además, ofrece servicios al Grupo ICE (ICE, RACSA, CNFL), de calibración y asesoramiento en métodos para toma de mediciones y en la elección de equipos de medición. Estos servicios se ofrecen a otras empresas e instituciones públicas y privadas; en el ámbito de su competencia y bajo estrictos parámetros que aseguren un alto índice de calidad.

La función que desempeña el Laboratorio de Metrología es esencial por tratarse de métodos eficaces para la toma de mediciones, selección de equipos y calibración de los mismos. Por tanto, las empresas estatales y privadas que contratan estos servicios, desean que sus equipos de medición posean una confiabilidad máxima; además de encontrarse dentro de los estándares nacionales.

Se extrae así, la importancia del Laboratorio Metrológico, ya que los servicios de calibración y medición que realiza, sirven como referencia a instrumentos de medición y calibración de otras instituciones y empresas. En la actualidad, el departamento se encuentra dirigido por el Dr. Harold Sánchez, jefe y encargado de la

implementación y desarrollo de las actividades destinadas por dicha dependencia. A este Laboratorio, se le ha encomendado la custodia de los patrones metrológicos y la realización de las calibraciones y ensayos pertinentes para asegurar una difusión fiel y veraz de la Metrología en el país; en el ámbito que le compete.

Algunos de los patrones más importantes establecidos como patrones nacionales; que se miden en el Laboratorio Metrológico son los de tiempo, corriente, voltaje, factor de potencia y nivel de frecuencia. Por ejemplo, el patrón primario de voltaje del ICE (declarado patrón nacional) consta de 4 módulos Fluke 732B, cada uno con una salida de  $10 V_{CD}$ . Su control estadístico es indispensable para asegurar su buen funcionamiento, en lo que a comparación de voltaje a nivel nacional se refiere.

El patrón nacional de tiempo, se mide por medio del servicio que brinda el NIST; el cual se realiza por medio de señales GPS (señales de posicionamiento global), de satélites de este Instituto usados para medir un total de cinco estándares de frecuencia a nivel mundial en forma simultánea. En el ICE, lo anterior se realiza por medio de la instalación de un sistema de medición de frecuencia (propiedad del NIST), el cual mediante GPS es monitoreado desde su central en los Estados Unidos; y por medio de una línea de teléfono dedicada, envía los resultados al sistema instalado en el Laboratorio Metrológico, desplegándolo en pantalla e imprimiendo periódicamente los mismos. Entre las actividades de esta dependencia, se pueden mencionar las siguientes:

- Calibración de instrumentos y equipos de medición con trazabilidad NIST.
- Caracterización de instrumentos y equipos de medición.
- Asesoramiento en la caracterización metrológica de mediciones y en la selección de equipos de medición.

Para realizar estas labores el departamento cuenta con el equipo suficiente y adecuado entre otros: patrones metrológicos, computadoras, acceso a Internet,

multímetros, osciloscopios y otros, los cuales sirven de ayuda para tal propósito.

## **1.2 Definición del problema y su importancia**

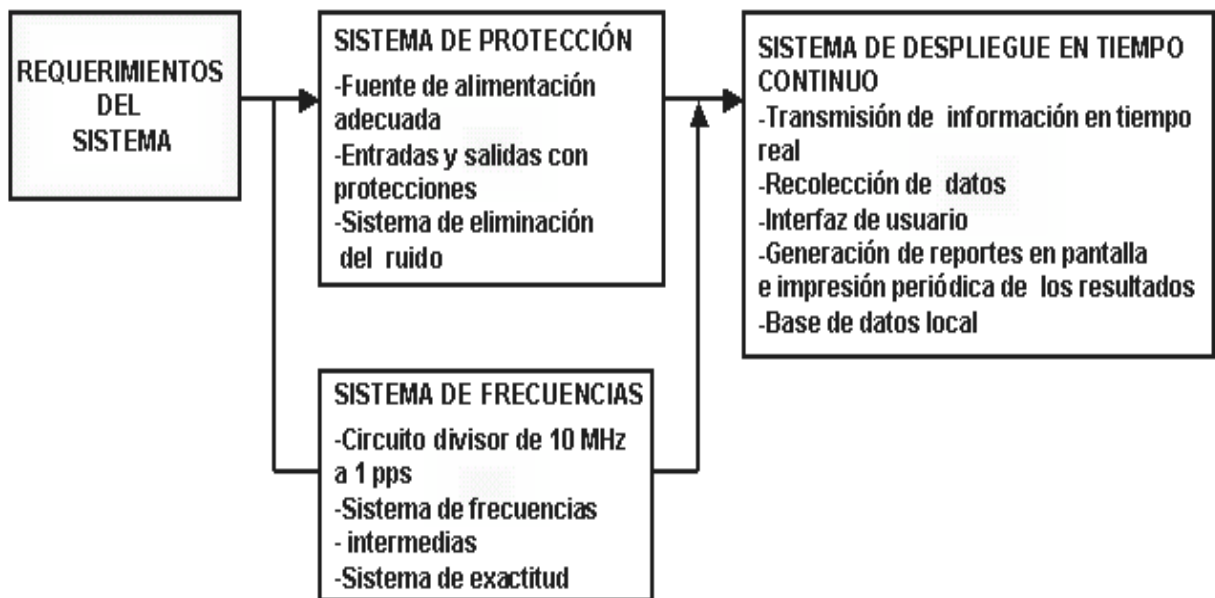
Dentro del Laboratorio Metrológico del ICE se mide la posición de un sistema de medición de frecuencia (propiedad del NIST), el cual mediante GPS se monitorea desde su central en los Estados Unidos; y por medio de una línea de teléfono exclusiva, envía los resultados de medición de frecuencia y tiempo. La posición de este sistema proporciona un patrón de tiempo, el cual sirve para establecer la hora nacional.

El problema principal de este sistema es su costo, pues aunque el sistema es eficiente; los costos por instalación, mantenimiento y prestación del servicio de medición de frecuencia, por parte del NIST son muy elevados. Si el sistema falla no se puede reparar a nivel nacional y debe ser sacado del país, o que un técnico especializado del NIST venga a repararlo, elevando costos. En consecuencia, en el ICE surgió la necesidad de investigar en un sistema alternativo que cumpliera con características similares a las del NIST. Con un sistema local para establecer la hora nacional, el ICE tendrá un patrón de referencia eficaz, por lo que los costos por medición de frecuencia y tiempo disminuirán sustancialmente, además del mantenimiento que será dado localmente. El ICE seguirá utilizando el servicio del NIST, pero en una modalidad distinta a la que actualmente tiene, disminuyendo costos.

Es importante señalar que el proyecto de implementar un patrón de tiempo, requiere del desarrollo de varias etapas, por lo que la primera etapa del proyecto final es la que se desarrolló en este proyecto de graduación. Se puede concluir, que uno de los beneficios más importantes del proyecto final será el de disminuir los costos por concepto de medición de tiempo y frecuencia en un 75% del costo original.

En consecuencia, se desarrolló una versión preliminar de un sistema que servirá como el patrón nacional de tiempo; una importante referencia para muchos de

los sistemas de medición de otras instituciones y empresas, públicas y privadas, además del ICE. Así, se diseñó e implementó un sistema que dentro del Laboratorio de Metrología, pudiera establecer por sí mismo la hora nacional, con recursos que se poseen en el país. El sistema se desarrolló con una señal procedente de un oscilador de cesio de 10 MHz, la cual pasó por divisiones sucesivas, hasta obtener una señal de un pulso por segundo que establecerá la hora nacional. El resultado se despliega en pantalla utilizando una PC, mediante una interfaz amigable para el usuario. Se diseñaron circuitos de protección del sistema, para evitar daños en los dispositivos involucrados en los procesos de detección y procesamiento de señales; y se disminuyeron señales indeseables como el ruido. El diagrama del sistema de integración requerido se observa en la figura 1.



POWERPOINT

**Figura 1** Requerimientos del sistema de integración.

El sistema de monitoreo se implementó en tiempo real. Es importante resaltar que se diseñó una base de datos o cuadro comparativo mediante el cual se pudieran observar las variaciones para cada uno de los patrones, tanto del NIST como el

generado por el ICE, teniendo como opción la impresión de los mismos. Sin embargo, dado que este proyecto se desarrolló en una etapa preliminar, sólo se trabajó con la señal proveniente del oscilador de cesio de 10 MHz, por lo que no se realizó la respectiva comparación con la señal proveniente del NIST; a pesar de que se dejó previsto dentro del programa, hacerlo en etapas posteriores del mismo. Lo explicado anteriormente afectó los porcentajes de error obtenidos en este proyecto (Ver figura 5.7), debido a que como se muestra en la figura, al no tener la señal proveniente del NIST, no se está realizando la comparación con la señal proveniente del oscilador de cesio. En consecuencia, el porcentaje de error es de cero por ciento pues no se está dando la comparación.

Además de crear el cuadro comparativo con la señal proveniente del oscilador de cesio y una futura señal proveniente del NIST, se generó un sistema de frecuencias intermedias. Este sistema de frecuencias intermedias fue implementado por sugerencia del ICE, con la finalidad de utilizar a mediano plazo estas señales para diferentes propósitos, como por ejemplo calibrar ciertos equipos.



## **1.3 Objetivos**

### **1.3.1 Objetivo general**

Diseñar, desarrollar e implementar, un prototipo para establecer la hora nacional en forma local y automática, a partir de una señal de un oscilador de cesio de 10MHz; mediante la síntesis de dispositivos y programas de computadora en un plazo de 16 semanas.

### **1.3.2 Objetivos específicos**

Para lograr el objetivo general de este proyecto fue necesario llevar a cabo los siguientes objetivos específicos:

1. Diseñar un módulo preliminar para la adquisición y manejo de datos o señales, mediante el uso de una tarjeta (protoboard); que divida una señal de 10 MHz proveniente de un generador de frecuencia en un pulso por segundo y genere frecuencias intermedias.
2. Implementar circuitos de protección en la tarjeta interfaz ISA (protoboard) para la adquisición de datos contra sobrevoltajes.
3. Desarrollar circuitos para la eliminación de ruido en la tarjeta interfaz ISA (protoboard).
4. Ensamblar el circuito anterior (tarjeta protoboard de interfaz con bus ISA, circuitos contra sobrevoltajes y ruido) que es el módulo de adquisición de datos, para someterlo a pruebas de funcionamiento y rendimiento.
5. Realizar pruebas de funcionamiento y eficiencia a la tarjeta de adquisición de datos.

6. Implementar una aplicación de software en bajo nivel que realice todo el manejo de las interrupciones del reloj interno de la PC, manejo del direccionamiento de la tarjeta y procesamiento de la información.
7. Establecer el programa en lenguaje de alto nivel, que maneje de forma eficiente las interrupciones realizadas en el lenguaje de bajo nivel.
8. Diseñar una aplicación en un lenguaje de alto nivel capaz de actualizar la hora, corregida mediante la introducción de una interrupción generada para reducir el intervalo de cuenta de los Timers de la PC y así aumentar la precisión de la PC; al igual que el sensado de la señal por la tarjeta de adquisición de datos.
9. Desplegar en la pantalla de una PC la hora a partir de la señal de 10 MHz mediante la creación de una interfaz amigable al usuario.
10. Crear dentro de la aplicación un cuadro comparativo de forma que deje previsto observar las mediciones obtenidas por el sistema diseñado (o señal del oscilador de cesio) y las medidas por una señal proveniente del NIST (en el caso de que una señal del NIST sea conectada al sistema en etapas posteriores a este proyecto), estableciendo los porcentajes de error respectivos.
11. Imprimir los resultados en intervalos de tiempo definidos en la aplicación en alto nivel creada.
12. Realizar pruebas en el laboratorio (pruebas de fallos) para verificar el programa de software (en alto y bajo nivel) que se encarga del control de la tarjeta o módulo de adquisición de datos que interconecta la PC con la señal.
13. Realizar pruebas de funcionamiento general para comprobar si el funcionamiento del sistema responde a los requerimientos establecidos.

14. Realizar pruebas de exactitud con todas las partes del proyecto ensambladas y corregir eventuales errores que se presenten con la evaluación.
15. Preparar la documentación para la transferencia del conocimiento tecnológico adquirido durante la ejecución del proyecto.

## **CAPÍTULO 2** **ANTECEDENTES**

---

### **2.1 Estudio del problema a resolver**

La necesidad de investigar sobre un sistema alternativo que realice funciones similares a la del NIST, fue una inquietud planteada este año por el Dr. Harold Sánchez, jefe y encargado de la implementación y desarrollo de las actividades del Laboratorio Metrológico del ICE. La idea surge por el deseo de actualizar los sistemas de control de toma de medidas en tiempo real y de forma local en el país, mediante la aplicación de la electrónica en la búsqueda de este propósito; además de tener una alternativa de medición confiable, que pueda ser modificada de acuerdo a los requerimientos futuros del Laboratorio Metrológico.

La idea del proyecto comenzó cuando se buscó una solución para monitorear y establecer la hora nacional. Se propuso diseñar un sistema usando recursos nacionales, que definiera patrones en tiempo real. En la actualidad existe un programa creado por el NIST, que proporciona una comparación entre el patrón de tiempo medido por un sistema previamente instalado dentro del Laboratorio Metrológico (propiedad del NIST) y el patrón de tiempo de la estación base del NIST ubicada en Colorado, Estados Unidos.

Con un estándar de incertidumbre relativa de  $5 \times 10^{-13}$  se obtienen las calibraciones de frecuencia, utilizando el servicio de medición de frecuencia del NIST (FMS). El FMS, usa señales del Sistema de Posicionamiento Global (GPS); es decir, señales de satélites que son tomadas como referencia de frecuencia. El sistema incluye un receptor de GPS y todo el hardware y software necesarios para automatizar el proceso de calibración. El ICE como suscriptor de este servicio, recibe un servicio completo del sistema de medición de frecuencia; donde se instala un equipo en el Laboratorio de Metrología, propiedad del NIST, como se explicó anteriormente. La posición de este sistema proporciona un patrón de tiempo, el cual sirve para establecer la hora nacional. Asimismo, este patrón se compara con el

patrón del satélite de la NIST por medio de las señales del sistema GPS; estableciendo el porcentaje de error entre ellos. Los resultados comparativos son impresos en intervalos específicos durante el día a través de una impresora conectada a una PC que recibe las señales, actualizando de esta forma el patrón de tiempo.

El sistema puede simultáneamente medir cinco estándares de frecuencia establecidos a nivel mundial; y desplegar en forma gráfica, en la pantalla de una PC e impresión, los resultados de los mismos. Es importante resaltar que en el ICE sólo se mide un patrón de frecuencia o estándar, correspondiente a Latinoamérica y específicamente, Centroamérica. El NIST provee además, el soporte necesario para el sistema. El cliente, en este caso el ICE, lo único que debe hacer es suplir los estándares de frecuencia a ser medidos y tener una línea de teléfono dedicada; la cual permite al personal del NIST, llamar a cada sistema conectado para verificar y analizar los datos. De esta forma, rápidamente se puede resolver los posibles problemas que surjan.

El servicio automatizado de tiempo (ACTS) permite que el módem de la computadora donde se usa el programa, se sincronice con el reloj del teléfono. En intervalos específicos durante el día, mediante la conexión a una PC de una impresora, el programa imprime un reporte. Este reporte proporciona el porcentaje de error entre los patrones de señales medidos en ambos satélites con una incertidumbre relativa de  $5 \times 10^{-13}$ . El sistema esta dedicado en forma exclusiva a esta labor y tiene un costo anual aproximado de \$6000.

El sistema del NIST es un sistema que cumple con normas internacionales, por lo que si tiene algún tipo de falla debe ser solucionada por técnicos especializados. Además, los costos por concepto de prestación de servicios (anualidad) son elevados, así como los correspondientes al transporte y la reparación del equipo. Otra desventaja es que el equipo debe ser sacado del país para su reparación, lo cual interrumpe el establecimiento del patrón de tiempo en el país. Lo anterior, debido a que el equipo del NIST es el único que brinda un patrón confiable de tiempo. En consecuencia, se hace imprescindible diseñar e implementar un

sistema alternativo, el cual brinde en caso de que el sistema del NIST tenga alguna falla, un patrón de tiempo confiable.

## **2.2 Requerimientos de la empresa**

El ICE pretende generar una propuesta alternativa que se adapte a la realidad nacional y que sea en la medida de lo posible implementada con los recursos que se poseen en el país. Así, generar un sistema que satisfaga las características de calidad, fiabilidad, precisión y facilidades de manejo; capaz de competir con sistemas de medición similares (NIST). A nivel nacional no había una propuesta definida ni diseños previos. Por lo tanto, se definió que el sistema debía ser automático e integrado a los existentes.

Partiendo de éstas premisas, el ICE sugirió desarrollar un sistema capaz de generar un patrón de tiempo por medio de un oscilador de cesio de 10 MHz, que estuviera interconectado con el bus ISA de la PC (esto por características de velocidad y capacidad de aislar el ruido de una mejor forma); y que por medio de un programa desarrollado en un lenguaje de generación anterior (sugerido C++) se desplegaran los datos en pantalla. El sistema tendría además, la posibilidad de manejo estadístico de los datos por medio de una base de datos creados para tal fin. Dentro de las consideraciones propuestas se formularon posibles circuitos de protección del sistema, para así evitar daños en los dispositivos usados. Además, se disminuyó la presencia de señales indeseables como el ruido; mediante el uso de técnicas adecuadas de construcción y selección de circuitos para este fin. El ruido dentro de este proyecto es crítico, debido a que si se presenta un alto índice de éste, se pueden ver seriamente afectados los resultados.

Es importante señalar que una de las características sugeridas para el proyecto fue la creación de una interfaz gráfica con el usuario, por lo que se sugirió la utilización del lenguaje C++. Sin embargo, este lenguaje no se usó y la aplicación fue programada en Delphi 5 por su flexibilidad en el manejo de ventanas dentro de la aplicación. Se consideró importante poder tener un control total del flujo de datos de la tarjeta ISA desde la misma aplicación; es decir, el usuario debería tener la

oportunidad de visualizar la hora, acceder desde el programa a la base de datos (o cuadro comparativo), diagnosticar el funcionamiento del bus ISA (en el caso de que el programa no estuviera funcionando) y poder invocar una ventana de ayuda con diversos temas sobre la aplicación actual. El manejo gráfico de ventanas en C++ es muy complejo, por lo que hace que los programas sean más difíciles de probar y depurar; y más complejos de adaptar a requisitos cambiantes. En consecuencia, por los requerimientos anteriormente mencionados y las facilidades gráficas de Delphi 5 se programó la interfaz en este lenguaje.

La señal utilizada proviene de un oscilador de cesio equivalente a 10 MHz. La unidad del sistema desarrollado es el segundo (s), como unidad de tiempo del Sistema Internacional de Unidades (SI). Un segundo es la duración de 9 192 631 770 períodos de la radiación asociada a la transición hiperfina del estado base del átomo de cesio 133. Lo anterior sirve como criterio de selección de los osciladores; porque la precisión del oscilador será proporcional a los períodos de transición del oscilador.

El cesio se escogió como generador de tiempo por sus características de estabilidad y precisión; además de ser el propuesto por el ICE. Otros tipos de osciladores usados en procesos de medición de tiempo y frecuencia, son los de cuarzo y rubidio. Los mejores osciladores son el rubidio y el cesio, debido a su alta precisión. El oscilador de cesio conserva las características de estabilidad originales, por un período de tiempo más largo que el de rubidio. La incertidumbre de un oscilador de cesio es de  $\pm 1 \times 10^{-12}$  (con un nivel de confianza de aproximadamente 95%); mientras que la de un oscilador de rubidio es de  $\pm 5 \times 10^{-11}$ ; lo cual demuestra la escogencia para establecer la hora patrón al oscilador de cesio.

## 2.3 Solución Propuesta

El ICE planteó las razones que motivaron la realización del proyecto. De esta forma, se visualizó en forma básica la solución, la cual consistió en un módulo independiente de adquisición y división de señal con conexión a una PC, mediante el uso de una tarjeta insertable y compatible con el estándar escogido ( en el caso de este proyecto, el estándar ISA) de la PC. Para esto se utilizó una PC en el Laboratorio Metrológico; y se implementó una tarjeta para el estándar ISA. De esta forma fue reconocida por el sistema operativo de la computadora al ser insertada en el slot. Esta solución se formuló con la intención de crear una mayor velocidad en lo que al tratamiento de la señal proveniente del oscilador de cesio, se refiere.

Se desarrolló dentro del módulo de adquisición de datos (compuesto por la tarjeta protoboard ISA, el microcontrolador, los circuitos de protección y ruido), un programa para el microcontrolador, el cual fue el encargado del manejo de la señal proveniente del oscilador de cesio. Dentro de la estructura del microcontrolador (la cual será explicada más adelante en el capítulo 4) se escogió para la división de las frecuencias un registro interno de éste llamado registro de opciones, el cual se programa en forma directa para tener intervalos de distintas frecuencias, según los requerimientos del sistema. Asimismo, se utilizaron los timers internos del microcontrolador para dividir la frecuencia y mantener el funcionamiento del oscilador del sistema. En este sentido, se escogió el Timer1 para controlar el reloj en tiempo real. El Timer1 fue escogido porque como se explicará el capítulo 4 tiene su propio cristal, el cual permite que el módulo funcione durante una operación llamada sleep (cuando el sistema entra en un modo de consumo mínimo de energía).

Se implementó un módulo de recepción de datos, el cual tuvo como función primordial interpretar la información recibida para generar los despliegues en pantalla, por medio de la aplicación de un “software” de alto nivel que se encargó del manejo de la información enviada por la tarjeta de adquisición, es decir, creó el direccionamiento de la misma. De esta forma, a través de la creación de un enlace



entre el lenguaje de bajo nivel con uno de alto nivel y la creación de una interfaz de usuario, se procesó la información y se pudo desplegar en pantalla. El lenguaje de programación utilizado fue Delphi 5; y fue seleccionado por su capacidad y flexibilidad de crear uniones o comunicaciones con el lenguaje de bajo nivel. Es importante destacar que este lenguaje ofrece la opción de utilizar ciertos componentes, creados para una función específica. En el caso de este proyecto, se utilizó un componente que lee directamente en bus ISA; situación que facilitó la programación de la interfaz en alto nivel.

Debido a que los tiempos de toma de medidas por parte de la PC deben ser muy precisos, para optimizarlos, además de su refrescamiento, se trabajó también con el timer de la PC. Para el manejo de los temporizadores encargados de este refrescamiento, se hizo uso de enlaces entre los lenguajes de bajo y alto nivel; controlando así, la interrupción que gobierna los temporizadores internos de la PC.

En consecuencia, el sistema de toma de datos (pulsos de la señal de 10 MHz, previamente dividida) fue ajustado, para poder obtener una precisión mayor a la ofrecida por la computadora (si se desea hacerlo). A continuación se hace una descripción detallada de los requerimientos del sistema:

- Comunicación con el módulo de adquisición de datos y divisor de frecuencia mediante una tarjeta compatible con el estándar PCI o ISA.
- Capacidad de alimentar todos los dispositivos involucrados con una única fuente de alimentación.
- Entradas y salidas con protecciones.
- Poseer circuitos de protección de fuente ante eventual daño ocasionado por sobrevoltajes en la red.
- Salidas intermedias de frecuencias (si es posible).
- Programa con facilidades de manejo (amigable para el de usuario), con el fin de tener un manejo estadístico de los patrones de tiempo de la señal del oscilador de cesio, en comparación con la del NIST.

Es importante destacar que el enlace creado entre bajo y alto nivel fue una interrupción, cuyo objetivo es el de responder a la petición de transmisión de datos de un dispositivo de hardware, en este caso el microcontrolador. De esta forma, se creó un vector de interrupción que contiene la dirección de esta función. En el sistema utilizado para este proyecto el primer Kbyte de memoria (desde 00000H a 003FFH) es utilizado como una tabla de vectores de interrupción. Para apuntar a cualquier dirección del mapa de memoria son necesarios cuatro bytes, 16 bits para el desplazamiento y 16 bits para el segmento. Luego, un Kbyte de memoria permite almacenar 256 vectores de interrupción. Algunos de los 256 vectores de interrupción son utilizados por el sistema, otros están libres para ser usados por los programas de usuario.

El enlace instaló una rutina de interrupción en el canal de interrupción IRQ1, que es el reloj del sistema. Este timer generó una interrupción 18.2 veces por segundo, el cual es el tiempo establecido por las interrupciones al reloj del computador. En la rutina de servicio de la interrupción se incrementa una variable global. Cuando esta variable es igual a 18 se presenta en la pantalla (como resultado del número de interrupciones por segundo).

## **CAPÍTULO 3**

### **PROCEDIMIENTO METODOLÓGICO**

---

#### **3.1 Procedimiento metodológico**

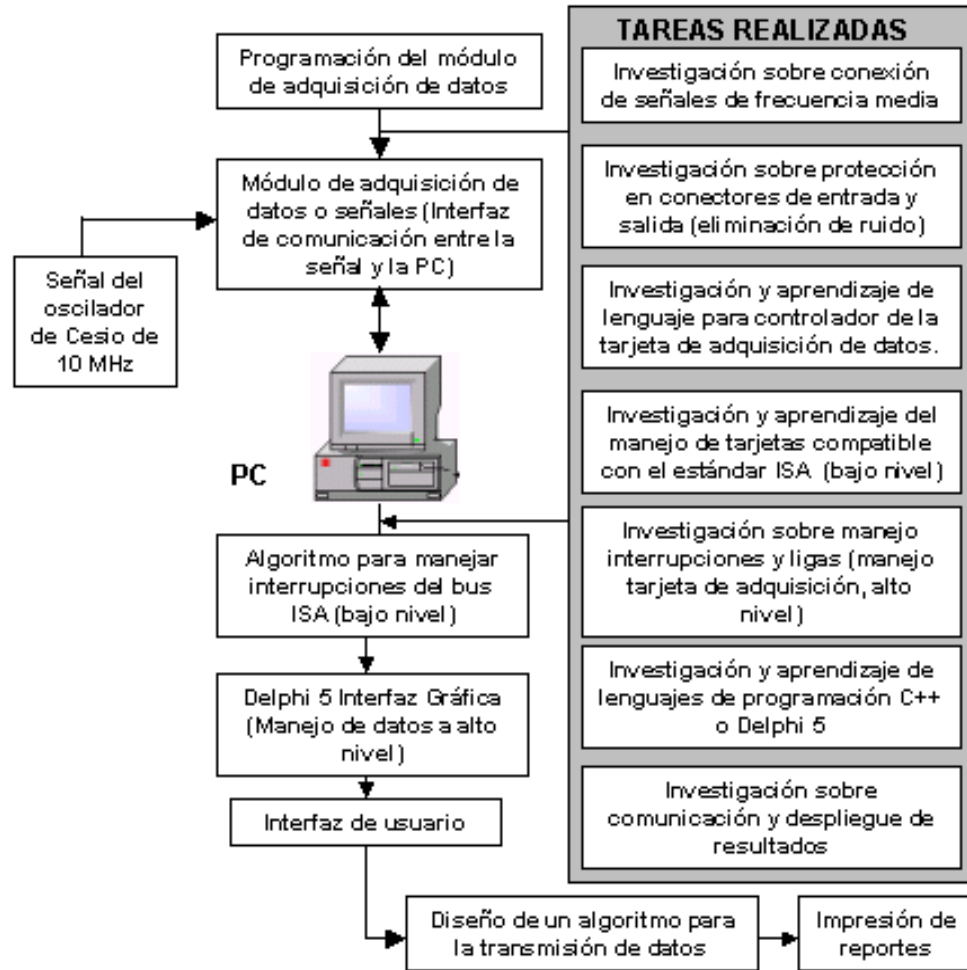
En la figura 2, se observan las principales tareas realizadas en este proyecto para solucionar el problema. En el bloque gris denominado “*tareas realizadas*”, se tiene la etapa de investigación y aprendizaje de tecnología para cumplir con los objetivos; como lo son los lenguajes de programación (software, hardware, enlaces entre alto y bajo nivel), aspectos técnicos la confección y manejo de la tarjeta de adquisición de datos, etc. Una vez obtenidos estos puntos, se contó con un criterio más amplio para desarrollar las tareas de los bloques que se conectan a la PC.

Asimismo, se aplicaron los conocimientos en ingeniería en el desarrollo de algoritmos para la transmisión y toma de información procedente de la tarjeta en forma directa y correcta; y la unión de los sistemas individuales en uno integrado, entre otros. Se realizó una investigación exhaustiva para determinar el mejor dispositivo para dividir la frecuencia; y para escoger el lenguaje en alto nivel con el cual desarrollar la interfaz con el usuario.

El módulo de adquisición de datos, tuvo como principal componente al microcontrolador 16F877, el cual fue el encargado de dividir una señal de entrada de 10 MHz. Para cumplir esta función, el microcontrolador (PIC, en lo sucesivo) fue programado utilizando un lenguaje de bajo nivel llamado MPLAB. Dentro de éste, se programó una rutina para obtener además de la señal de un pulso por segundo (señal que establece el patrón de tiempo), frecuencias intermedias que serán utilizadas posteriormente para calibrar otros instrumentos dentro del Laboratorio Metrológico que así lo requieran.

La interfaz gráfica (lenguaje alto nivel), fue realizada en Delphi 5. Entre las rutinas que se programaron, están la de lectura del bus ISA, la creación de una base de datos y la opción de impresión de resultados. Es importante resaltar que la base de datos se despliega en pantalla si el usuario así lo desea. Se crea además, un

archivo de texto que puede ser cargado en Excel o desde WordPad, mismo que se actualiza en forma periódica y paralela a la lectura de datos en el programa. Lo anterior, se explicará con mayor detalle en el capítulo 5.



VISIO TECHNICAL4.1

**Figura 2** Tareas realizadas para obtener la solución.

Para obtener una solución adecuada, la cual satisficiera los objetivos propuestos en el capítulo anterior, se procedió de la siguiente forma:

1. Para diseñar un módulo preliminar para la adquisición y manejo de datos o señales, mediante el uso de una tarjeta (protoboard) que tomara una señal de 10 MHz proveniente de un generador de frecuencia y la dividiera en un pulso por segundo; además de generar frecuencias intermedias, se realizó lo siguiente:

**a)** Se examinó las consideraciones necesarias para poder hacer uso del estándar PCI o ISA de la PC, que garantizaran una aplicación segura y efectiva en procesos de comunicación. Para lo anterior se hizo uso de Internet, libros acerca del tema, etc.

Se escogió el bus ISA por ser uno de los estándares con mayor velocidad; y principalmente por estar dentro de la computadora (lo cual disminuía la influencia de ruido dentro del sistema). Otro aspecto que determinó la escogencia del bus ISA fue del de manejar un bus de direcciones de 20 bits y un bus de datos de 8 bits, el cual permitía trabajar con la mayoría de las señales de interrupción del PC, e incluso utilizar los circuitos de DMA (si fuera necesario).

**b)** Se investigó y estableció el mejor microcontrolador para realizar la división de frecuencia proveniente del oscilador de cesio. Así se escogió el microcontrolador 16F877 de Microchip, el cual tiene un registro donde se programa directamente la división de la frecuencia, disminuyendo el retardo que pudiera presentarse en este proceso. Otra característica importante que se estudió para elegir finalmente al controlador PIC16F877, fue la de la rapidez para realizar cálculos en un tiempo limitado y la precisión de los datos por manejar.

**c)** Una vez seleccionado el microcontrolador, se procedió a estudiar a profundidad las características del microcontrolador seleccionado y del software para programarlo. Se usó una PC para leer las hojas de datos del fabricante y el manual del software de programación, una vez instalado. Entre las características estudiadas para el PIC16F877, se mencionan las siguientes: arquitectura, segmentación, la arquitectura basada en bancos de registros, formato de instrucciones, juego de instrucciones, instrucciones ortogonales y las herramientas de soporte, etc.

**d)** Al conocer el lenguaje de programación del microcontrolador y su set de instrucciones, se procedió a programar el PIC por medio de software. Dentro de su programación, se contempló el generar como salidas del microcontrolador, frecuencias intermedias. Para lo anterior, se usó de una PC a disposición para tal fin.

**2.** Implementar circuitos de protección en el módulo preliminar (protoboard) de adquisición de datos contra sobrevoltajes, tuvo como paso metodológico:

Investigar y establecer el mejor circuito de protección contra sobrevoltajes. Fue necesario además, estudiar las posibles causas de ocurrencia de sobrevoltajes e implementar las soluciones más factibles ante eventualidades de ese tipo; mediante el uso de Internet, libros acerca del tema, etc. Al investigar sobre este tema se escogió un regulador de voltaje LM7805, cuya circuitería interna permite detectar cualquier nivel de voltaje incorrecto, brindando seguridad al resto del sistema. Si ésta rebasa el valor de la corriente permitida (Ver anexo 2), se limitará automáticamente la corriente de salida hasta que se elimina la sobrecarga.

**3.** Para desarrollar circuitos contra el ruido en el módulo de adquisición de datos preliminar (protoboard), se procedió de la siguiente forma:

**a)** Se investigó y estableció el mejor circuito contra ruido (Ver en el apéndice A.2, los diodos Zener conectados a las entradas de los BNC), en el mayor grado posible; haciendo uso de Internet, libros acerca del tema, etc. Se definió, que por costo y abundancia en el mercado nacional se trabajaría con diodos Zener, pues no sólo poseen la característica de regular el voltaje en las entradas de la frecuencia (entrada de señal de 10 MHz), sino que también muestran una baja tolerancia al ruido. Por lo tanto, una señal con dos diodos Zener conectados presentará un nivel muy bajo de ruido como podrá observarse en el capítulo de resultados.

**b)** Establecidos los componentes del módulo de adquisición (microcontrolador, circuitos de protección y contra ruido, resistencias, etc), se procedió a su encargo. Es importante destacar que se compraron los componentes del exterior.

**4.** Ensamblar el circuito anterior que fue el módulo de adquisición de datos, para someterlo a pruebas de funcionamiento y rendimiento siguió el siguiente orden:

**a)** Se procedió a programar el PIC con el programador correspondiente. Para lo anterior, se usó una PC a disposición y el respectivo programador. Además, se hicieron pruebas para comprobar el buen funcionamiento del programa dentro del PIC. Entre las pruebas realizadas (ver capítulo de resultados), se midió por medio de un osciloscopio las diferentes señales tanto de entrada como de salida, para comprobar el funcionamiento del proceso de división de frecuencia programado en el PIC.

**b)** Con los componentes pedidos a disposición, se procedió a alambrear y conectar los diferentes componentes del módulo de prueba (protoboard). Así, se conectó el microcontrolador (además de sus entradas y salidas), los circuitos de protección contra sobrevoltajes y ruido; y demás componentes. Se usó una protoboard o wire-wrap y las herramientas necesarias para el respectivo alambrado.

**c)** Una vez alambrado el módulo preliminar de adquisición de datos, se realizaron las pruebas respectivas sobre su funcionamiento y eficiencia; utilizando para generar la señal de 10 MHz, un generador de frecuencias. Se probaron otros aspectos como la influencia de ruido, voltajes de alimentación adecuados en entradas y salidas, con protecciones que funcionaran de forma segura y eficiente. Además del generador de frecuencias, se empleó multímetros, puntas de prueba, la PC (si acaso se tenía que cambiar la programación del PIC), etc.

**5.** Realizar pruebas de funcionamiento y eficiencia a la tarjeta (protoboard) de adquisición de datos, siguió el siguiente procedimiento:

Una vez listo el módulo final de adquisición de datos, se realizaron las pruebas respectivas sobre su funcionamiento y eficiencia; utilizando para generar la señal de 10 MHz un generador de frecuencias con este valor. Se probaron otros aspectos como la influencia de ruido, voltajes de alimentación adecuados en entradas y salidas. En la sección de resultados, se pueden observar varios oscilogramas al respecto. Además del generador de frecuencias, se utilizaron multímetros, puntas de prueba, etc. Un aspecto importante, fue el de modificar la programación del PIC, en

uno de sus módulos. El módulo modificado fue el registro de opciones ya que el microcontrolador presentaba un valor inferior a 1 Hz por segundo situación que incumplía con uno de los requerimientos del sistema.

**6.** Para implementar una aplicación de software en bajo nivel que realizara todo el manejo de las interrupciones del reloj interno de la PC, manejo del direccionamiento de la tarjeta y procesamiento de la información, se estableció los siguientes pasos metodológicos:

**a)** Se investigó y estableció el mejor algoritmo para manejar el reloj interno de la PC, de forma tal que al final de la creación del mismo, se pudiera modificar los intervalos de toma de mediciones. Para realizar este paso, se hizo uso de una PC (para programar el algoritmo), además de Internet y libros sobre cómo intervenir el timer interno de la PC. Es importante señalar, que la interrupción que se generó para la lectura del bus ISA hace que el reloj interno de la computadora se modifique. Asimismo, los algoritmos de programación del PIC, tienen una consecuencia directa en los tiempos de transmisión de datos, debido a que el proceso de oscilación generado por el microcontrolador será el encargado de generar los pulsos necesarios para que el conteo de tiempo se dé.

**b)** Una vez creado el procedimiento para modificar el timer de la PC, se procedió a crear el respectivo procedimiento (siempre en bajo nivel), para el direccionamiento de la tarjeta de adquisición de datos, usando una PC; así como Internet. Este procedimiento está contenido dentro de la misma interrupción del bus ISA. Dentro de la interrupción, cuando se ejecuta un ciclo de lectura desde la tarjeta hará que el microprocesador del PC coloque una señal en alto (ALE), la cual habilita las señales de direccionamiento.

**7.** Establecer el programa adecuado en el lenguaje de alto nivel, que manejara de forma eficiente las interrupciones realizadas en el lenguaje de bajo nivel, es decir, que generara un enlace entre estos dos lenguajes; requirió lo siguiente:



Investigar y establecer el mejor lenguaje en alto nivel, mediante el cual, a través de la creación de una aplicación que garantice un buen manejo de las interrupciones creadas por el lenguaje de bajo nivel. Para realizar este paso, se hizo uso de una PC, Internet y libros sobre el tema, etc. Se estableció el lenguaje Delphi 5 como el indicado para desarrollar la aplicación en alto nivel, dadas sus características de rapidez y manejo a nivel gráfico (ventanas).

**8.** Al diseñar una aplicación en un lenguaje de alto nivel capaz de actualizar la hora, corregida mediante la introducción de una interrupción generada para reducir el intervalo de cuenta de los Timers de la PC y así aumentar la precisión de la PC; al igual que el sensado de la señal por la tarjeta de adquisición de datos, se siguió el siguiente orden:

**a)** Una vez seleccionado el lenguaje de alto nivel a utilizar, se procedió a estudiar a profundidad las características del mismo; con el objetivo de crear una aplicación que sirviera de enlace entre el lenguaje de alto nivel y el programa realizado con el de bajo nivel. Se hizo uso de una PC para diseñar la aplicación, así como de libros especializados acerca del lenguaje y su estructura de programación. Es importante resaltar, que el lenguaje Delphi ofrece múltiples herramientas o componentes, los cuales se encuentran a disposición del programador. Los componentes están contenidos en una barra de herramientas sobre el área de diseño. Cuando se inicializa diseñando una aplicación, el programa abre dos ventanas simultáneamente: una de código (donde se procederá a programar los componentes seleccionados) y una donde se colocaran los iconos de las plantillas que representan los componentes (ventana llamada forma gráfica). Las características de este lenguaje, permiten que un componente dentro de una plantilla sea seleccionado y colocado en una de las ventanas diseñadas para tal fin. El programador busca el componente deseado, arrastra el icono hacia la ventana de forma gráfica y da doble clic al icono. Lo anterior, hace que se abra la ventana de código y el programador pueda establecer

las rutinas de funcionamiento del componente seleccionado. Entre estos componentes, se encuentra uno que maneja la transferencia y lectura de datos del bus ISA, el cual se utilizó dentro de este proyecto.

**b)** Al tener las herramientas necesarias para crear la aplicación, se procedió a diseñar la misma.

**9.** Desplegar en la pantalla de una PC la hora a partir de la señal de 10 MHz mediante la creación de una interfaz amigable al usuario, incluyó el siguiente paso metodológico:

**a)** Establecido el programa de software se procedió a conectar a la entrada del módulo de adquisición de datos final, la señal proveniente de un generador de frecuencias de 10 MHz. De esta forma se verificó que tanto el programa el bajo nivel como el correspondiente en alto nivel y su respectivo enlace, funciones de forma tal que el resultado final fuera el despliegue en pantalla de la hora. Se usó un generador de frecuencias; además de multímetros, puntas de prueba, una PC, etc.

**10.** Para crear dentro de la aplicación, un cuadro comparativo de forma que deje previsto observar las mediciones obtenidas por el sistema diseñado (o señal del oscilador de cesio) y las medidas por una señal proveniente del NIST (en el caso de que una señal del NIST sea conectada al sistema en etapas posteriores a este proyecto), estableciendo los porcentajes de error respectivos; se requirió lo siguiente:

Funcionando el despliegue en pantalla se procedió a modificar la aplicación de software para establecer una pequeña base de datos (Ver figura 5.5 y 5.6). Lo anterior, con la finalidad de proporcionar un cuadro comparativo entre los dos patrones y generar el porcentaje de error obtenido a partir de esta comparación; además de su correspondiente despliegue en pantalla, como una opción si el usuario lo desea. Se hizo uso de una PC, Internet y libros. Es importante recalcar, que no se hizo la comparación con la señal proveniente del NIST, sino que se dejó previsto la

conexión de la misma; y que durante el desarrollo del proyecto no se trabajó con esta señal, sino con datos obtenidos a partir de la misma.

**11.** Imprimir los resultados en intervalos de tiempo definidos en la aplicación en alto nivel creada, siguió el paso metodológico detallado a continuación:

Se procedió a modificar la aplicación en alto nivel de forma tal que, si el usuario lo desea, pueda imprimir los resultados cuando lo desee y mediante la impresión del cuadro comparativo realizado y explicado en el apartado anterior; usando una PC, Internet y libros sobre el tema. La impresión se realiza en forma de un texto, ya que la base es guardada como un archivo de texto, con la flexibilidad de que puede ser abierto desde programas como Excel o Word.

**12.** Para realizar pruebas en el laboratorio (pruebas de fallos) para verificar el programa de software (en alto y bajo nivel) que se encarga del control de la tarjeta o módulo de adquisición de datos que interconecta la PC con la señal, se estableció el siguiente paso:

Se efectuaron las pruebas de manejo de datos, despliegue en pantalla e impresión; así como los tiempos para la toma de mediciones (relojes internos de la PC). Se hizo uso de una PC, el generador de frecuencias con una señal de 10 MHz y demás instrumentos para realizar las pruebas respectivas. En lo que respecta al cuadro comparativo (base de datos), se realizaron pruebas de toma de datos en diferentes intervalos de tiempos (entre 60 segundos y 60 minutos), de acuerdo a la escogencia del usuario en la opción, "*Configurar el tiempo de muestreo*" (Figura 5.7). Asimismo, se midieron los valores de las frecuencias intermedias generadas a partir del proceso de división de frecuencia, en especial la señal de salida de 1 Hz por segundo (muy importante en el momento en el cual se comparen los valores generados por el sistema desarrollado, con los de otros sistemas, como por ejemplo el NIST).

**13.** El paso metodológico para realizar pruebas de funcionamiento general para comprobar si el funcionamiento del sistema responde a los requerimientos establecidos, se explica a continuación:

Se realizaron las pruebas de manejo de datos (tanto en el módulo de adquisición como en la interfaz del usuario), despliegue en pantalla e impresión; así como los tiempos para la toma de mediciones (relojes internos de la PC), al conectarse la señal de 10 MHz proveniente del oscilador de cesio. Se hizo uso de una PC, el oscilador de cesio con una señal de 10 MHz y demás instrumentos para realizar las pruebas respectivas. Es importante señalar, que el despliegue en pantalla presentó un problema de retardo, el cual se explicará en el apartado de análisis de resultados.

**14.** Para realizar pruebas de exactitud con todas las partes del proyecto ensambladas y corregir eventuales errores que se presenten con la evaluación, se hizo lo siguiente:

Se hicieron pruebas de exactitud con todas las partes del proyecto ensambladas (módulo de adquisición de datos final, software en alto nivel o interfaz de usuario), para corregir eventuales errores que se presentaran con la evaluación y toma de mediciones. Además, se probó la eficiencia del sistema, los posibles retardos del mismo, el funcionamiento del timer de la PC, etc. Se realizaron pruebas con el osciloscopio para medir las señales de salida (producto de la programación del PIC); y pruebas en la transmisión de datos por el bus ISA (de forma tal que el proceso de oscilación generado por el algoritmo del PIC, pudiera establecer un patrón de tiempo confiable, similar al del NIST). Finalmente, se probó el sistema a nivel de software, es decir, el funcionamiento de la aplicación como tal. Se realizaron pruebas sobre el despliegue en pantalla, el funcionamiento del cuadro comparativo y las diferentes opciones contenidas dentro de la aplicación (configuración de tiempo de muestreo, impresión de resultados, almacenamiento de resultados en dispositivos alternos como un floppy u otro sector del disco duro en forma de archivo de texto, la ayuda

para el usuario del sistema desde la aplicación y detección del bus ISA dentro del inspector de recursos).

**15.** Preparar la documentación para la transferencia del conocimiento tecnológico adquirido durante la ejecución del proyecto, siguió los pasos metodológicos explicados a continuación:

**a)** Se reunió toda la información más relevante concerniente a la elaboración del módulo de adquisición de datos y los algoritmos de los lenguajes en alto y bajo nivel.

**b)** Se prepararon tanto los informes como la respectiva presentación del proyecto, con el análisis de resultados y sugerencias al respecto.

**c)** Como una opción para el Laboratorio Metrológico, se confeccionó un manual de usuario (guía rápida del usuario). En el mismo, se explican los métodos del diseño, el funcionamiento del mismo, las acciones a tomar en caso de que se requieran cambios al sistema, etc. Para esto se usó una PC con una presentación en PowerPoint, además de material impreso sobre el tema.

## **CAPÍTULO 4** **DESCRIPCIÓN DEL HARDWARE UTILIZADO**

---

### **4.1 Descripción del hardware utilizado**

#### **4.1.1 Equipo e Instrumentación**

##### **1. Computador:**

Para el desarrollo del proyecto fue utilizada una computadora. Las consideraciones mínimas requeridas consistieron en el que la PC tenga capacidad del manejo adecuado del bus ISA, soportado bajo el lenguaje de programación Delphi 5. Es importante dentro de un proyecto como este, hacer referencia a los relojes del computador o sus sistemas de tiempo. Los relojes son esenciales para el buen funcionamiento de cualquier sistema porque juegan un papel decisivo en la sincronización de procesos o la asignación de turnos de ejecución entre otras tareas relevantes realizadas por un computador.

En un computador, generalmente se cuenta con dos relojes en el sistema: uno que lleva la hora y fecha del sistema, además de oscilar entre 50 y 60 veces por segundo y el reloj que oscila entre 5 y 100 millones de veces por segundo y que se encarga de enviar interrupciones al CPU de manera periódica. El reloj de mayor frecuencia sirve para controlar el tiempo de ejecución de los procesos, para despertar los procesos que están 'durmiendo', entre otras actividades.

Para mantener la hora y fecha del sistema generalmente se usa un registro alimentado por una pila de alta duración que almacena estos datos. Así, aunque se suspenda la energía la fecha permanece. Para lanzar procesos (chequeo de tiempo ocioso de un dispositivo, por ejemplo), se almacena un valor en un registro (valor QUANTUM) el cual se decrementa con cada ciclo del reloj, y cuando llega a cero se dispara un proceso que ejecutará las operaciones necesarias (escoger un nuevo proceso en ejecución, verificar el funcionamiento del motor del disco flexible, hacer eco de un carácter del teclado, etc).

## 2. Timer de la PC: ( “CHIP” interno relacionado, 8253)

Otro de los aspectos importantes a considerar en los dispositivos de hardware con que se trabajó, es el relacionado con las rutinas que se encargan de manipular los “timers” internos de la PC. Lo anterior se realiza de forma tal que se dé una interrupción por software (rutina programada en la aplicación de alto nivel), que sustituya la interrupción original utilizada por la CPU en sus procesos de “timing” internos.

El timer de la PC, se incluye dentro de los dispositivos, debido a que uno de los aspectos fundamentales del proyecto fue el de lograr mayor precisión en la toma de datos que la ofrecida por la PC. Por ejemplo, para un sistema basado en los microprocesadores de Intel, la *INT8* o interrupción del timer del sistema, fue lo que se modificó para darle mayor resolución al cronómetro de la PC. Para esto se deben tener en cuenta varias consideraciones previas: una PC al igual que otros tipos de computadoras, poseen un *circuito controlador de interrupciones*, denominado **PIC**, que se encarga de estar chequeando continuamente las líneas de solicitud de interrupción (**IRQ**) que puedan ocurrir mientras se dedica a la realización de distintas tareas. La PC puede atender comúnmente una totalidad de 15 interrupciones donde se encuentra la siguiente asignación de acuerdo al IRQ solicitante:

**Tabla 4.1** Asignación de IRQ

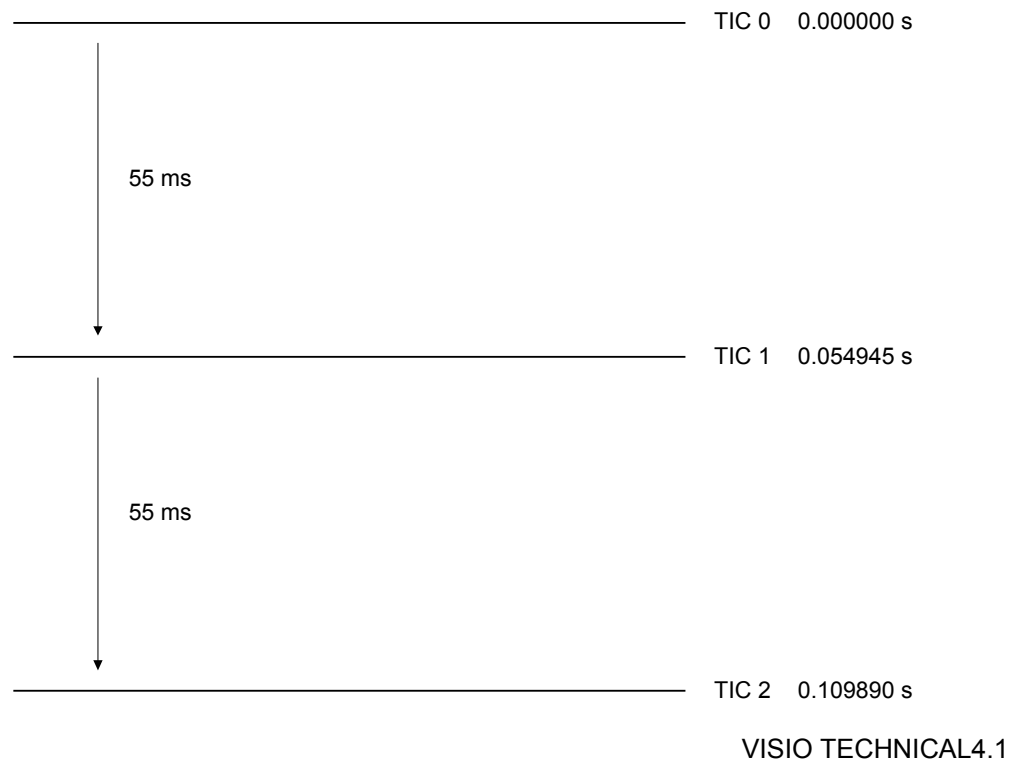
| <b>Número de interrupción</b> | <b>Acción</b>  |
|-------------------------------|--|
| 00                            | Cronómetro de Sistema  |
| 01                            | Teclado 101/102 teclas estándar o Microsoft Natural Keyboard |
| 02                            | Controlador Programable de Interrupciones                    |
| 03                            | Puerto de Comunicaciones (COM2)                              |
| 04                            | Puerto de Comunicaciones (COM1)                              |
| 05                            | SB16 Audio Device  |
| 06                            | Controlador estándar de disquetes                            |
| 07                            | Puerto de Impresora (LPT1)                                   |
| 08                            | Sistema CMOS/reloj en tiempo real                            |
| 09                            | External Midi (Mpu401) Device                                |
| 10                            | Windows Sound System (WSS) Device                            |
| 11                            | Intel 82371AB PCI to USB Universal Host Controllers          |
| 11                            | Marcador IRQ para manejo de PCI                              |
| 13                            | Procesador de datos numéricos                                |
| 14                            | Intel 82371AB Bus Master IDE Controller                      |
| 14                            | Controlador Primario IDE (FIFO única).                       |
| 15                            | Intel 82371AB PCI Bus Master IDE Controller                  |
| 15                            | Controlador Secundario IDE (FIFO única)                      |

EXCEL

Las funciones de “timing” externo son realizadas en la computadora por medio del dispositivo conocido como Programmable Interval Timers (PIT8253), que se encarga de dichas funciones mediante programación. Este se encuentra conformado por tres contadores conocidos como “counter 0”, “counter 1” y “counter 2”; encargados de llevar, el “Reloj de Tiempo Real”, el control de los tiempos del refrescamiento de la memoria RAM (en conjunto con el “chip” 8237 (Controlador Programable de Acceso Directo a Memoria DMA)), y la generación de tonos en el “speaker” de la PC, junto con el “chip” 8255 (Programmable Peripheral Interface) como contador en los procesos de manejo de unidades de disco flexible.

La PC cuenta con una resolución, por defecto, de  $55ms$ ; es decir un refrescamiento de tiempo de 18.2 veces por segundo, de acuerdo a las posibilidades del generador de reloj que alimenta los “timers”. A cada cuenta del timer se le conoce como TIC de reloj, por lo tanto, si se tomaran muestras entre cada TIC de reloj consecutivos, éstas se diferenciarían entre sí en  $55ms$  (ver figura 4.1).





**Figura 4.1** Precisión ofrecida por defecto por la PC, en la toma de muestras consecutivas.

### 3. Estándar ISA

El bus ISA que se observa en la figura 4.2 maneja un bus de direcciones de 20 bits y un bus de datos de 8 bits. Permite trabajar con la mayoría de las señales de interrupción del PC, e incluso utilizar los circuitos de DMA (Direct Memory Access). Las principales señales del bus ISA se muestran en la tabla 4.2. Todas las señales del bus ISA emplean niveles TTL estándar. A continuación se explica su funcionamiento con un ciclo de lectura desde un puerto de entrada/salida. Lo primero que hace el microprocesador es poner la señal ALE hasta un nivel alto, entonces envía la dirección del puerto a través de las señales A0-A19. Después, la señal ALE vuelve a nivel bajo. En adelante la dirección del puerto a ser leído quedará retenida en un latch. Entonces el bus pone IOR a nivel bajo. El dispositivo direccionado enviará un byte de datos a través de las líneas D0-D7 del bus de datos. El microprocesador leerá el bus de datos y pondrá la señal IOR a nivel alto de nuevo.

| SIGNAL NAME | PIN NUMBER  | SIGNAL NAME |
|-------------|-------------|-------------|
|             | GND B1 A1   | -I/O CH CK  |
| +RESET DRV  |             | +D7         |
| +5V         |             | +D6         |
| +IRQ2       |             | +D5         |
| -5V         |             | +D4         |
| +DRQ2       |             | +D3         |
| -12V        |             | +D2         |
| RESERVED    |             | +D1         |
| +12V        |             | +D0         |
|             | GND B10 A10 | -I/O CH RDY |
| -MEMW       |             | +AEN        |
| -MEMR       |             | +A19        |
| -IOW        |             | +A18        |
| -IOR        |             | +A17        |
| -DACK3      |             | +A16        |
| +DRQ3       |             | +A15        |
| -DACK1      |             | +A14        |
| +DRQ1       |             | +A13        |
| -DACK0      |             | +A12        |
| CLOCK       | B20 A20     | +A11        |
| +IRQ7       |             | +A10        |
| +IRQ6       |             | +A9         |
| +IRQ5       |             | +A8         |
| +IRQ4       |             | +A7         |
| +IRQ3       |             | +A6         |
| -DACK2      |             | +A5         |
| +T/C        |             | +A4         |
| +ALE        |             | +A3         |
| +5V         |             | +A2         |
| +OSC        |             | +A1         |
| GND         | B31 A31     | +A0         |

VISIO TECHNICAL4.1

Figura 4.2 Configuración interna del bus ISA.

Un ciclo de lectura a un puerto funciona de la siguiente manera: El microprocesador pone la señal ALE a "1", entonces envía la dirección del puerto a través de A0-A19. ALE es puesta a nivel bajo. El microprocesador envía el byte de datos que será escrito. Luego pone un "0" en IOW. Después que el dispositivo ha tenido tiempo de leer el byte, el microprocesador pone la señal IOW a nivel alto de nuevo. La única diferencia entre un ciclo de lectura/escritura a memoria y un ciclo de lectura/escritura a un puerto consiste en que en un ciclo de memoria se utilizaran las señales MEMR y MEMW de la misma manera que se hace con IOR y IOW.

**Tabla 4.2** Listado de las principales señales para el bus ISA

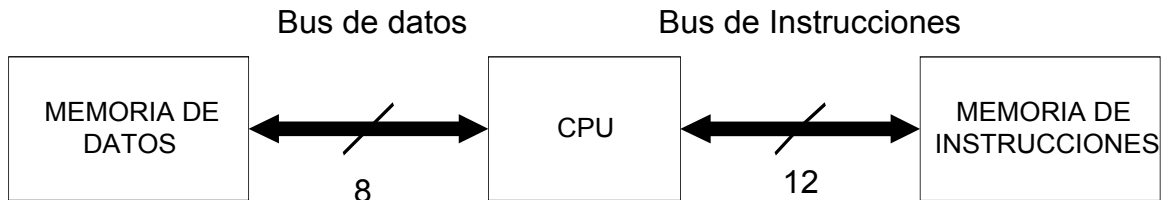
| SEÑAL                    | DESCRIPCION   |
|--------------------------|---|
| A0-A19 (pines A31 a A12) | Bits de dirección 0-19, permiten direccionar 1Mb de memoria y 64K de puertos de e/s.  |
| AEN                      | Address Enable; Cuando está activa el controlador DMA posee el control de las líneas de dirección y del BUS de datos, conforme se indique en MEMR/MEMW. Cuando está inactiva la CPU tiene el control de estas líneas.         |
| ALE                      | Address Latch Enable (salida); se emplea para que la CPU esté aislada de las líneas de dirección (triestado). Es forzado activado durante los ciclos DMA.   |
| CLK                      | Señal de reloj del sistema (actual velocidad del BUS).  |
| D0-D7                    | Bits de datos 0-7 para e/s a memoria o puertos de e/s.  |
| DACK0-DACK3              | Reconocimiento DMA para los canales 0 al 3; empleada por el controlador para reconocer una petición DMA (validación de acceso DMA). DACK0 es empleada para el refresco de memoria (MREF).                                     |
| DRQ1-DRQ3                | Petición DMA 0-3; empleada por periféricos que desean los servicios del controlador DMA; Se mantiene activa hasta que la correspondiente señal DACKx se hace activa.  |
| IOR                      | I/O Read; indica a un dispositivo de e/s que coloque su dato en el BUS del sistema.   |
| IOW                      | I/O Write; indica a un dispositivo de e/s a leer un dato del BUS del sistema.   |
| IRQ2-IRQ7                | Petición de interrupción 2-7; indica a la CPU que un dispositivo de e/s necesita servicio.  |
| MEMR                     | Memory Read; esta señal es producida por la CPU o el controlador DMA e indica a la memoria que debe introducir el dato direccionado en el BUS del sistema. Presente tanto en el BUS PC como en la extensión AT.               |
| MEMW                     | Memory Write; esta señal es producida por la CPU o el controlador DMA e indica a la memoria que debe leer y almacenar el dato presente en el BUS. Presente tanto en el BUS PC como en la extensión AT.                        |
| OSC                      | Oscilador; Señal de reloj de 14.31818 MHZ (periodo de 70ns); 50% del ciclo de servicio.   |
| RESET DRV                | Reset Drive; empleada para resetear la lógica del sistema.  |
| TC                       | Terminal Count; produce un impulso cuando la cuenta final de un canal DMA es alcanzado. Asimismo, el controlador de DMA activa esta señal para hacer saber a un periférico que el número programado de bytes ha sido enviado. |

EXCEL

#### 4. El microcontrolador PIC16F877:

Se utilizó este dispositivo por tener una velocidad de ejecución de instrucciones muy alta, además de poseer un registro programable para dividir la

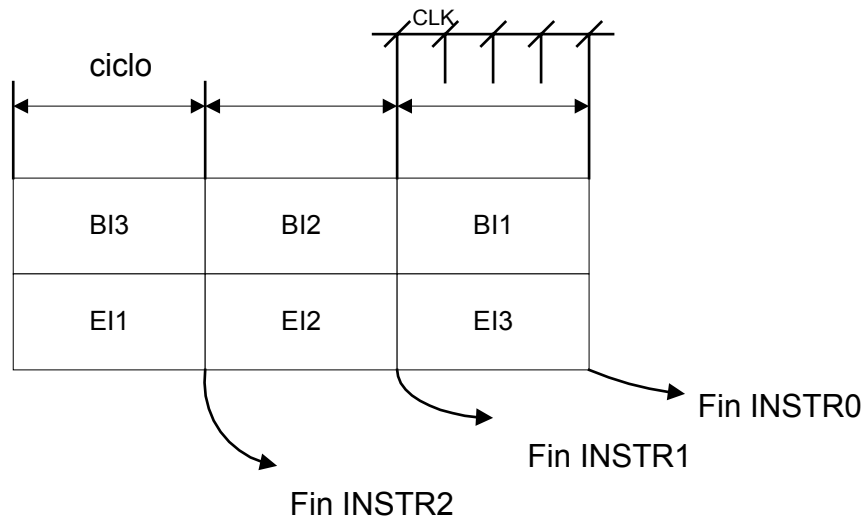
frecuencia que se aplique a cualquiera de sus entradas. Asimismo, los microcontroladores trabajan con la arquitectura Harvard y aplican la técnica de segmentación (pipe-line) en la ejecución de instrucciones. En la arquitectura Harvard, el CPU se conecta de forma independiente y con buses distintos con la memoria de instrucciones y con la de datos, como se observa en la figura 4.3.



VISIO TECHNICAL4.1

**Figura 4.3 Arquitectura Harvard para el microcontrolador 16F877**

La arquitectura Harvard permite que el CPU acceda simultáneamente a las dos memorias. La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj). Lo anterior se puede observar en la figura 4.4.



VISIO TECHNICAL4.1

**Figura 4.4 Segmentación en el microcontrolador 16F877**

La segmentación permite al procesador ejecutar cada instrucción en un ciclo de instrucción equivalente a cuatro ciclos de reloj. En cada ciclo se realiza la búsqueda de una instrucción y la ejecución de la anterior.

En el microcontrolador 16F877, el formato de todas las instrucciones tiene la misma longitud. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones; y facilita enormemente la construcción de ensambladores y compiladores. Asimismo, todas las instrucciones son ortogonales; es decir, cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

En lo que respecta al reloj del microcontrolador, este dispone de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema. Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C. Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

#### ***4.1 Recursos especiales del PIC 16F877***

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. En el caso específico de este proyecto, se utilizó este tipo de microcontrolador por incorporar las siguientes características:

- Temporizadores o “Timers”.

- Perro guardián o “Watchdog”.
- Protección ante fallo de alimentación o “Brownout”.
- Estado de reposo o de bajo consumo.
- Conversor A/D.
- Conversor D/A.
- Comparador analógico.
- Modulador de anchura de impulsos o PWM.
- Puertas de E/S digitales.
- Puertas de comunicación.

#### 4.1.1 *Temporizadores o “Timers”*

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempos se carga un registro con el valor adecuado y a continuación, dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las entradas o salidas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

#### 4.1.2 *Perro guardián o “Watchdog”*

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continua. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema. Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al

Perro guardián y, al completar su temporización emitirá una señal, la cual provocará el reset.

#### *4.1.3 Protección ante fallo de alimentación o “Brownout”*

Se trata de un circuito que produce un reset al microcontrolador cuando el voltaje de alimentación ( $V_{DD}$ ) es inferior a un voltaje mínimo (“brownout”). Mientras el voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene en reset, comenzando a funcionar normalmente cuando sobrepase dicho valor.

#### *4.1.4 Estado de reposo ó de bajo consumo*

Son abundantes las situaciones en las cuales el microcontrolador debe esperar, sin realizar ninguna tarea, hasta que se produzca algún acontecimiento externo que lo ponga nuevamente en funcionamiento. Para ahorrar energía, los microcontroladores disponen de una instrucción especial (SLEEP ), que los pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se “congelan” sus circuitos asociados, quedando el microcontrolador en un estado de reposo. Sin embargo, si dentro del algoritmo de programación existe alguna tarea en la cual el microcontrolador esté realizando constantemente (como por ejemplo, el proceso de división), el estado de reposo del mismo se da sin interrumpirla. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

#### *4.1.5 Conversor A/D (CAD)*

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las entradas del circuito integrado.

#### *4.1.6 Conversor D/A (CDA)*

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que emite al exterior por una de las salidas de la cápsula.

#### *4.1.7 Comparador analógico*

Algunos modelos de microcontroladores disponen internamente de un amplificador operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las entradas del encapsulado. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra. También hay modelos de microcontroladores con un módulo de tensión de referencia que proporcionan diversas tensiones de referencia que se pueden aplicar en los comparadores.

#### *4.1.8 Modulador de anchura de impulsos o PWM*

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las entradas del integrado.

#### *4.1.9 Puertas de E/S digitales*

Todos los microcontroladores destinan algunas de sus entradas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando compuertas. Las líneas digitales de las compuertas pueden configurarse como entrada o como salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.



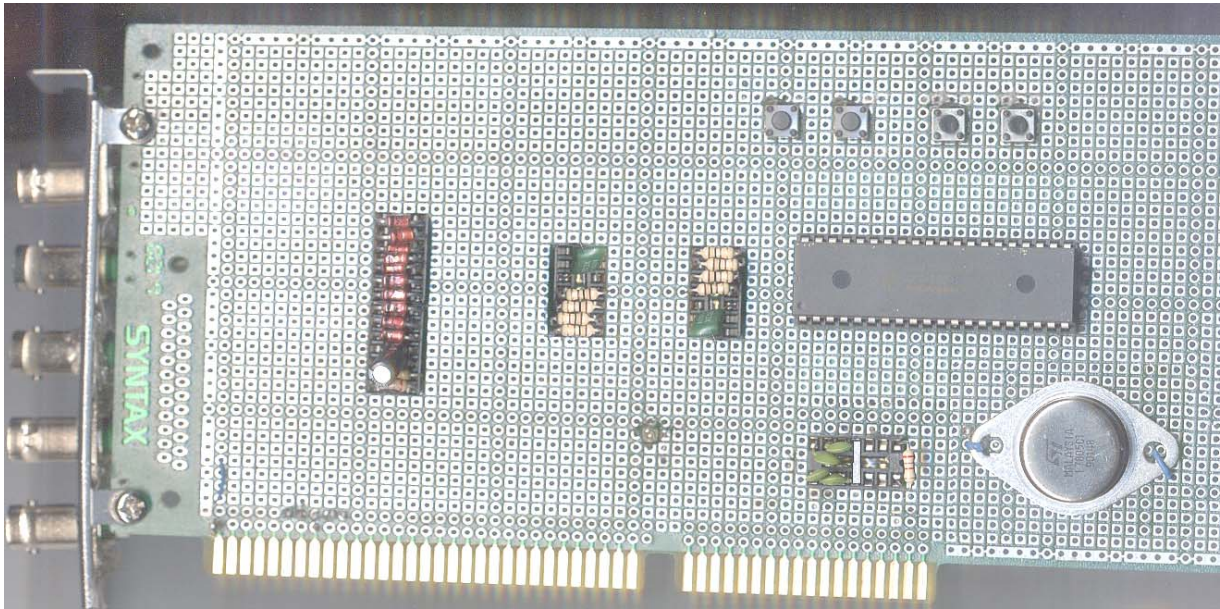
#### 4.1.10 Puertas de comunicación

Con objetivo de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. El PIC16F877 dispone de recursos que permiten directamente esta tarea, entre los que destacan:

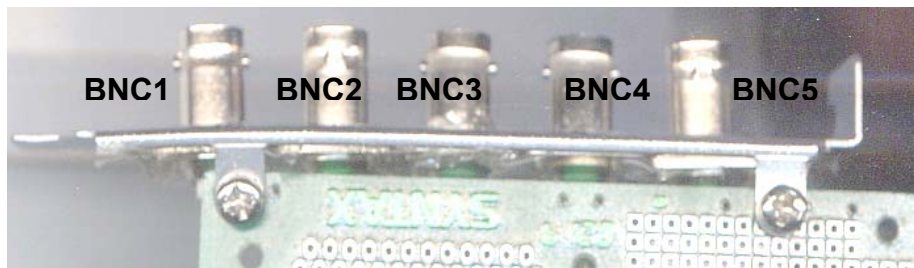
- UART, adaptador de comunicación serie asíncrona.
- USART, adaptador de comunicación serie síncrona y asíncrona
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- USB (Universal Serial Bus), que es un moderno bus serie para los PC.
- Bus I<sup>2</sup>C, que es un interfaz serie de dos hilos desarrollado por Phillips.
- CAN (Controller Area Network), para permitir la adaptación con redes de conexión multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos.

#### 5. Módulo de adquisición de datos o señal:

Esta tarjeta consiste en un módulo que interconecta el bus ISA de la computadora con la señal proveniente del oscilador de cesio de 10 MHz. La tarjeta de interconexión (la cual se observa en la figura 4.5 y 4.6), representa el medio mediante el cual la PC logra la adquisición de la señal. Esta se encuentra conformada por un circuito limitador de voltaje (ver apéndice A.2), que como su nombre lo indica, limita la fuente de alimentación que se conecte a la interfaz, a un voltaje de 5 V, ya que los circuitos que la conforman son de tecnología TTL.



**Figura 4.5** Fotografía de módulo de adquisición y transmisión de datos (protoboard). Vista superior

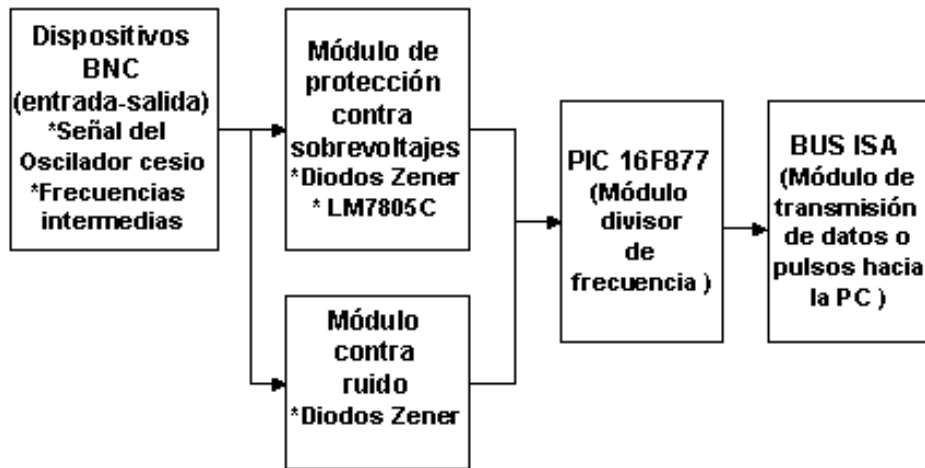


**Figura 4.6** Fotografía de módulo de adquisición y transmisión de datos (protoboard). Vista lateral (conectores BNC)

La tarjeta de adquisición de datos está conformada por los módulos que se observan en la figura 4.7. Los diferentes módulos y su funcionamiento se detallan a continuación:

- a) Dispositivos de entrada-salida (BNC): Los dispositivos BNC, son los encargados de recibir la señal del oscilador de cesio de 10MHz y transmitirla hacia el PIC 16F877. Es importante recalcar que como se observa en la figura 4.6, la tarjeta de adquisición de datos posee 5 conectores BNC, de los cuales el conector BNC1 será el encargado de

recibir la señal proveniente del oscilador de cesio; los conectores BNC2, BNC3 y BNC4 emiten las frecuencias intermedias. Finalmente, el conector BNC5 se encargará de emitir una señal de 1 Hz. Esta señal será un patrón de comparación entre el sistema desarrollado a partir del oscilador de cesio y otros sistemas que funcionen de manera similar (como es el caso del sistema que provee el NIST).



VISIO TECHNICAL 4.1

**Figura 4.7** Diagrama de bloques para el módulo de adquisición y transmisión de datos (protoboard).

- b) **Módulo de protección contra sobrevoltajes:** Este módulo está formado por un regulador de voltaje LM7805, el cual limita cualquier pico de corriente dentro de la tarjeta a un valor seguro. Este valor se encuentra dentro de la llamada área de protección (Ver anexo 2), la cual hace que las salidas de los transistores internos del regulador se provean de un límite de potencia interna de disipación. Esto protege en forma directa al módulo de división de frecuencia o microcontrolador. El LM7805 es un dispositivo de tres terminales, con un rango de voltaje de 5V y una corriente de salida de hasta 1 A. Como se explicó anteriormente, los circuitos internos de este dispositivo detectan una corriente de carga. Si ésta rebasa un valor determinado, se limita automáticamente la corriente de salida hasta que se

elimina la sobrecarga. Asimismo, este regulador se encargará de medir la diferencia del voltaje y de la corriente de entrada y salida, para comprobar que no exista ninguna combinación no permitida. Si existe este tipo de problema, el regulador se apaga. Esta característica es conocida como *protección de área segura* (Ver anexo 2).

Los diodos Zener cumplen una función de protección y aislamiento de ruido, en las terminales de entrada y salida de los BNC. Como los diodos Zener está diseñado para que funcione en la zona de ruptura, pueden mantener una tensión constante con independencia de que se presenten grandes variaciones de la tensión de la red (señal por el BNC) o de la resistencia de carga. En condiciones normales, el diodo Zener debe tener polarización inversa y además de trabajar en la zona Zener la tensión de la fuente  $V_S$  debe ser mayor a la tensión ruptura  $V_Z$ . Por esta razón se emplea resistencias en serie (Ver apéndice A.2), cuya función es la de limitar la corriente a un valor menor de su limitación máxima de corriente, pues si no se realiza esta conexión, el diodo Zener se puede quemar como cualquier dispositivo que disipase excesiva potencia.

- c) Módulo contra ruido: Este módulo esta compuesto por los diodos Zener los cuales no sólo limitan los voltajes y las corrientes mayores a las permitidas, sino que hacen que señales indeseables de ruido no sean introducidas en la tarjeta. Los diodos Zener en este caso, actúan como un filtro mediante el cual, las señales de ruido (pequeñas) van a ser disipadas por el par de diodos conectados en las terminales del BNC. Se utilizó también para disminuir el nivel de ruido, cable coaxial especial para el manejo de señales de osciloscopios; el cual se obtuvo en el Laboratorio de Electrónica del ICE San Pedro. El cable coaxial se conectó entre la entrada de los BNC y las entradas y salidas del microcontrolador, lo que disminuyó también el nivel del ruido. En el capítulo de resultados se muestran dos oscilogramas en los cuales se demuestra lo anterior. Asimismo, el PIC 16F877 fue seleccionado por las bajas señales de ruido que pueden

generarse dentro de él. Al ser un circuito integrado, su circuitería interna, además de su encapsulado permite que haya un bajo nivel de ruido ya que no se requiere interconectar otros dispositivos externos, su alambrado es poco y la señal que proviene del oscilador de cesio ya ha sido filtrada.

- d) Módulo oscilador y divisor de frecuencia (PIC 16F877): Este módulo esta conformado por el PIC 16F877, el cual será el encargado de dividir la señal del reloj de cesio de 10 MHz hasta un pulso por segundo. Asimismo, se encargará de direccionar esta señal hacia el bus ISA. El módulo de adquisición de datos tuvo la particularidad de direccionar usando un puerto hacia al bus ISA, la señal dividida; además de proporcionar en las salidas tres de los puertos del PIC, las frecuencias intermedias requeridas. Una exigencia de este sistema es la regulación estricta de los tiempos que duran las diversas acciones que realiza. El dispositivo típico destinado a gobernar los tiempos recibe el nombre de temporizador o "timer" y, básicamente, consiste en un contador ascendente o descendente que determina un tiempo entre el valor que se le carga y el momento en que se produce su desbordamiento o paso por 0. Los microcontroladores PIC tienen como mínimo dos temporizadores. Uno de ellos actúa como principal y sobre él recae el control de tiempos de las operaciones del sistema. El otro recibe el nombre de perro guardián o "Watchdog".

El Watchdog vigila que el programa no presente dificultades en su funcionamiento, es decir que no detenga su funcionamiento por un espacio de tiempo prolongado; y dejen de ejecutarse las instrucciones secuenciales que ha previsto el diseñador. Para realizar esta labor de vigilancia, el watchdog cada cierto tiempo, realiza un proceso en donde comprueba si las funciones programadas se están ejecutando adecuadamente en el CPU; en caso contrario, por ejemplo si el control está detenido en un bucle infinito o a la espera de algún acontecimiento no se produce, el watchdog se activa y provoca un reset, reiniciando todo el sistema.

Tanto el temporizador principal (TMR0), como el perro guardián (WDT) precisan controlar tiempos largos y aumentar la duración de los impulsos de reloj, que los incrementan o decrementan. Para cubrir esta necesidad, se dispone de un circuito programable llamado divisor de frecuencia que divide la frecuencia utilizada por diversos rangos para poder realizar temporizaciones más largas.

Para regular el comportamiento del temporizador principal, el watchdog y el divisor de frecuencia emplean algunos bits de la palabra de configuración y del registro de opciones (OPTION). El divisor de frecuencia puede aplicarse a uno de los dos temporizadores, al TMR0 o al WDT. Con el temporizador principal actúa en forma prioritaria, es decir, los impulsos pasan primero por el divisor de frecuencia y una vez aumentada la duración de los últimos, se aplican a TMR0. Sin embargo, el microcontrolador también puede actuar como divisor previo o "Prescaler", es decir, con el watchdog el divisor de frecuencia actúa después ("Post-scaler"). El divisor de frecuencia puede actuar al ritmo de una señal externa aplicada en la entrada T0CKI, o bien, con la señal de reloj interna del microcontrolador CLKOUT, procedente del oscilador propio. Mediante algunos bits del registro de opciones y la palabra de configuración, se controla el trabajo del divisor de frecuencia sobre el TMR0 o el WDT.

e) Módulo de transmisión de datos o pulsos hacia el computador (bus ISA): Es el encargado de recibir la señal dividida del PIC, así como otras señales de datos que serán las encargadas junto con esta de iniciar el proceso de oscilación y generar la interrupción para que se dé el despliegue de datos en pantalla desde la aplicación en alto nivel.

El circuito funciona de la siguiente forma:

- Una señal de 10 MHz se aplica a la entrada BNC, o entrada coaxial denominada J1 (ver apéndice A.2), la cual está conectada a la entrada

OSC1/CLKIN, la cual será la entrada del oscilador; misma que el microcontrolador verá como reloj externo para proceder a dividirla.

- El limitador de voltaje hace que no existan sobrevoltajes o algún tipo de cortocircuito en el módulo de adquisición de datos.
- La señal es dividida por el microcontrolador mediante el programa realizado en bajo nivel, el cual hará divisiones sucesivas hasta tener una señal de un pulso por segundo. Asimismo, como salidas se tienen tres frecuencias intermedias.
- Cuando la señal se divide hasta un pulso por segundo, este pulso es reconocido por la interfaz gráfica; y se inicia la etapa de software.

## **CAPÍTULO 5** **DESCRIPCIÓN DEL SOFTWARE UTILIZADO**

---

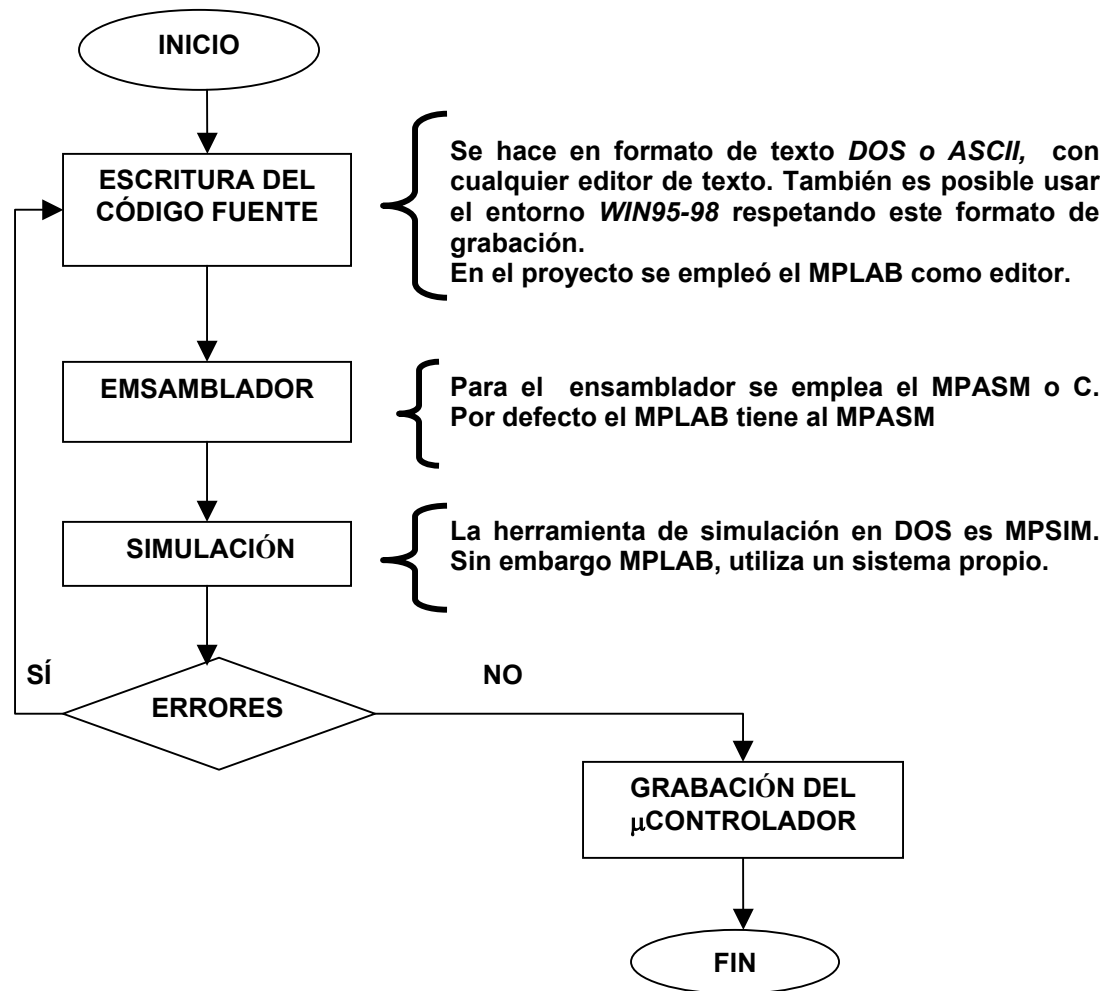
A nivel de software, el sistema del proyecto del reloj patrón se compuso de dos partes: la primera, el software desarrollado para programar el microcontrolador 16F877, en el lenguaje MPLAB de Microchip; y el software desarrollado en Delphi 5, para crear interrupciones en el timer de la PC de forma que alterara los tiempos de toma de mediciones, del módulo de adquisición de datos. Además, el programa en Delphi 5 fue el encargado de desplegar los datos en pantalla y crear la base de datos.

### **5.1 Descripción del software utilizado para programar el microcontrolador PIC16F877**

En cuanto a la programación del Microcontrolador PIC 16F877 se utilizó el MPLAB, que utiliza una estructura muy sencilla y rápida, la cual de muestra en la figura 5.1. En el programa preliminar se desarrolló una rutina para dividir la señal proveniente del oscilador de cesio de 10 MHz, haciendo uso de la programación de un registro interno del microcontrolador llamado registro de opciones. En el mismo, se programa la frecuencia de operación del microcontrolador, así como también se determina si la señal de frecuencia será externa, o bien se trabajará con su oscilador interno. En este mismo registro se programa el rango de división de la frecuencia.

La misión principal del registro de opciones es gobernar el comportamiento del temporizador principal. Su bit T0CS (Timer 0 External Clock Edge Select), elige en el multiplexor interno en el microcontrolador, la procedencia de los impulsos de reloj, que pueden ser los del oscilador externo ( $F_{osc}/4$ ) o los que se aplican desde el exterior por la pata T0CKI. El bit T0SE (Timer 0 Clock Source Select) selecciona el tipo de flanco que es activo para la frecuencia externa. Si T0SE = 1, el flanco activo es el descendente y si T0SE = 0, es el ascendente.

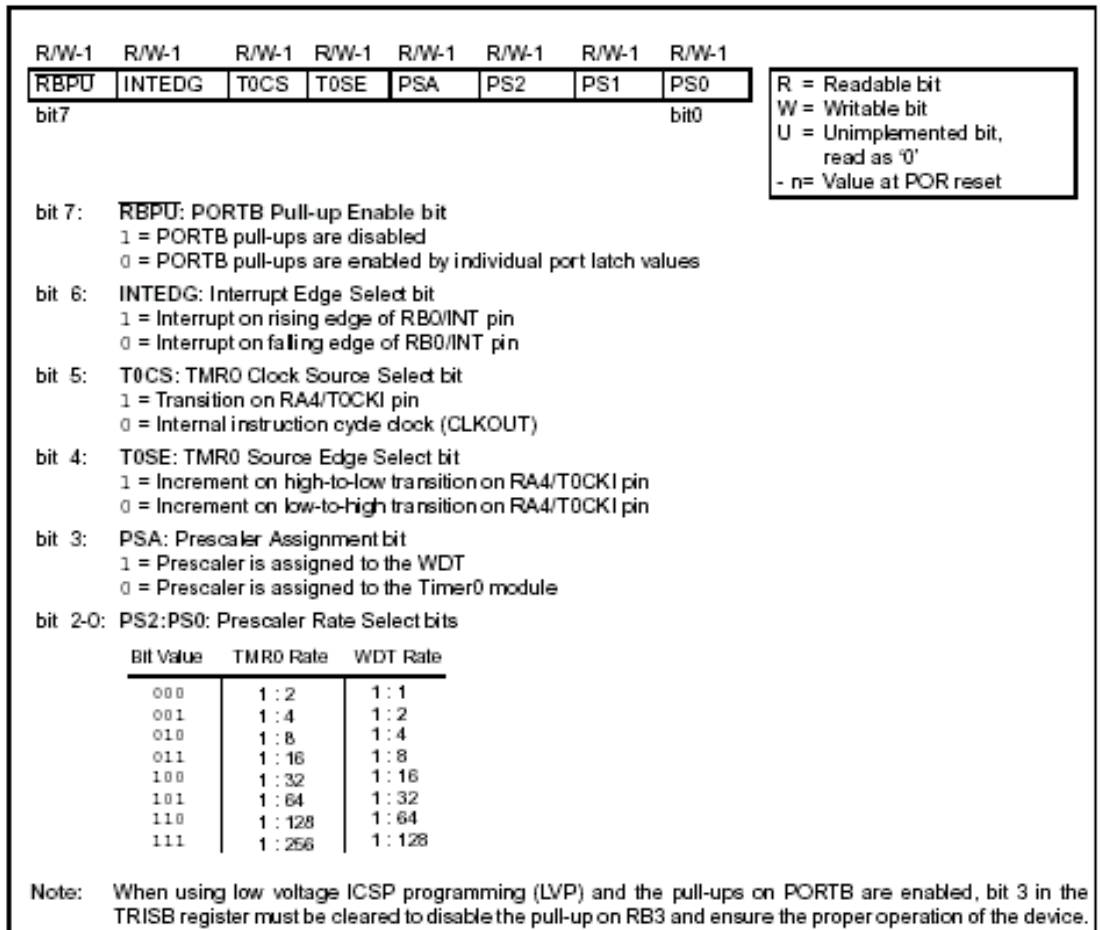




VISIO TECHNICAL 4.1

**Figura 5.1** Diagrama de desarrollo del MPLAB.

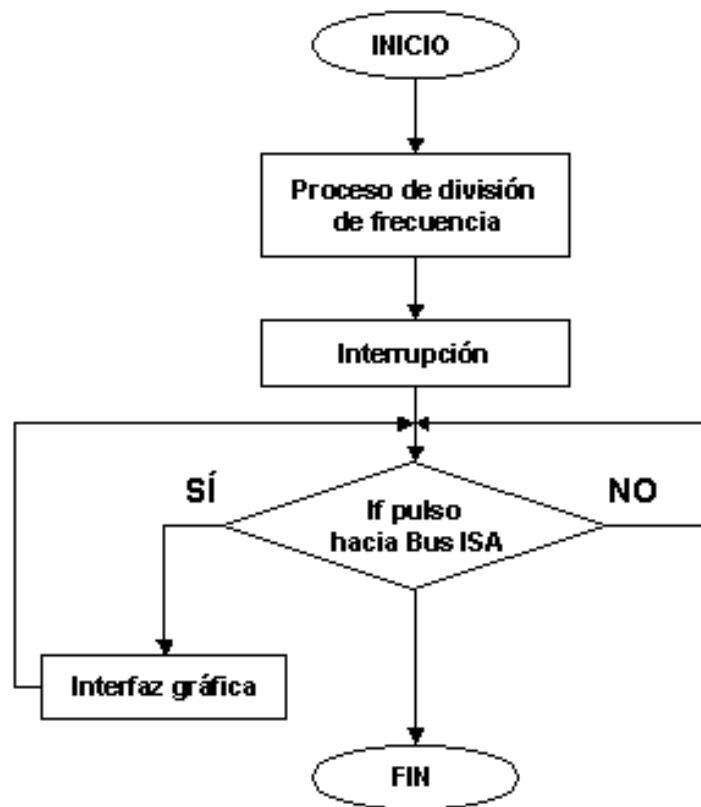
Como se observa en la figura 5.2, el bit PSA del registro de opciones tiene la función de asignar el divisor de frecuencia al TMR0 o al WDT. Si PSA = 0 la salida del visor se aplica a la patilla del microcontrolador TMR0; sin embargo si PSA = 1 el divisor se destina al perro guardián. Finalmente, los 3 bits de menos peso del registro de opciones (PSA2, PSA1 y A0) seleccionan el rango por el que el divisor de frecuencia va a dividir los impulsos que se le apliquen. En la figura 5.2 se ofrece la distribución y asignación de los bits en el registro de opciones.



HOJA DE DATOS 16F877 (MICROCHIP)

**Figura 5.2** Distribución y asignación de los bits del Registro de opciones. Los 6 bits de menos peso solo se pueden escribir (w), no hallándose implementados los dos de más peso, que siempre se leen como 0.

Dentro de la rutina del microcontrolador, también se programó el direccionamiento de los datos, tanto hacia la entrada de datos del bus ISA como hacia las salidas de las frecuencias intermedias a generar, el cual se muestra en el diagrama de flujo de la figura 5.3 El diagrama anterior muestra en forma general la introducción de la interrupción al programa, interrupción en la cual se estará actualizando el valor del pulso proveniente de la división de la señal del oscilador de cesio.



VISIO TECHNICAL 4.1

**Figura 5.3** Diagrama general para la programación del PIC 16F877

Para el desarrollo de las rutinas en bajo nivel se usó el MPLAB de Microchip. MPLAB es un conjunto de herramientas para el desarrollo y depuración de aplicaciones en un proyecto, que incluye un editor de texto, un manejador de proyectos para mantener el código organizado y un ambiente para depurar el software que se ha desarrollado. Este ambiente incluye el simulador MPLAB-SIM, y soporta otras herramientas de Microchip tales como el emulador PICMASTER y el programador de dispositivos PICSTART Plus. Cuando se le dice al sistema que corre en tiempo real, las instrucciones se ejecutan tan rápido como es posible por el software.

La velocidad del simulador depende de la velocidad de la computadora en la que se instale y de las demás tareas que se estén ejecutando al mismo tiempo que se ejecuta. El software debe de actualizar todos los registros en la RAM, también monitorear las entradas/salidas, ajustar y limpiar las banderas, revisar y buscar por

puntos de ruptura en el software; donde finalmente puedan visualizarse, las instrucciones de los microcontroladores tal y como están siendo ejecutadas en el CPU de la computadora.

## **5.2 Descripción del software de bajo nivel para generar la interrupción del bus ISA (lectura o transmisión de datos desde el bus hacia la aplicación de la PC)**

En el mapa de memoria del computador se pueden encontrar dos tipos de interrupciones, las de software y las de hardware. Las interrupciones externas son manejadas por el controlador de prioridad de interrupciones 8259A. Cuando el 8259A recibe una señal de interrupción a través de las señales IRQ2 a IRQ7, envía una señal de petición de interrupción a la entrada INTR del microprocesador. Entonces el microprocesador envía una señal INTA (Interrupt-Acknowledge) al 8259A. El microprocesador puede luego leer el tipo de interrupción del dispositivo externo. En el caso del sistema desarrollado, se usa el tipo de interrupción leído desde el dispositivo externo para obtener la dirección de la rutina de servicio de la interrupción de la tabla de vectores de interrupción en la memoria (Ver tabla 1). Aunque las señales INTR y INTA no están presentes en el bus ISA, estas señales solo son utilizadas por el microprocesador y el 8259A.

Lo explicado anteriormente se utiliza para programar las interrupciones. El lenguaje Delphi 5 posee un componente, el cual permite que la interrupción y lectura del bus ISA se realice de forma tal que no se tenga que programar externamente mediante otro lenguaje la misma. Este componente se inserta dentro de la aplicación y lo único que se debe programar es que debe realizar el programa con los datos que está recibiendo a partir de la lectura de ese bus. El objetivo básico del componente es generar una interrupción que ejecute una función que responda a la petición del dispositivo ISA. El vector interrupción contiene la dirección de esta función. En este sistema, el primer Kbyte de memoria (desde 00000H a 003FFH) es utilizado como una tabla de vectores de interrupción y para apuntar a cualquier dirección del mapa de memoria son necesarios cuatro bytes: 16 bits para el desplazamiento y 16 bits

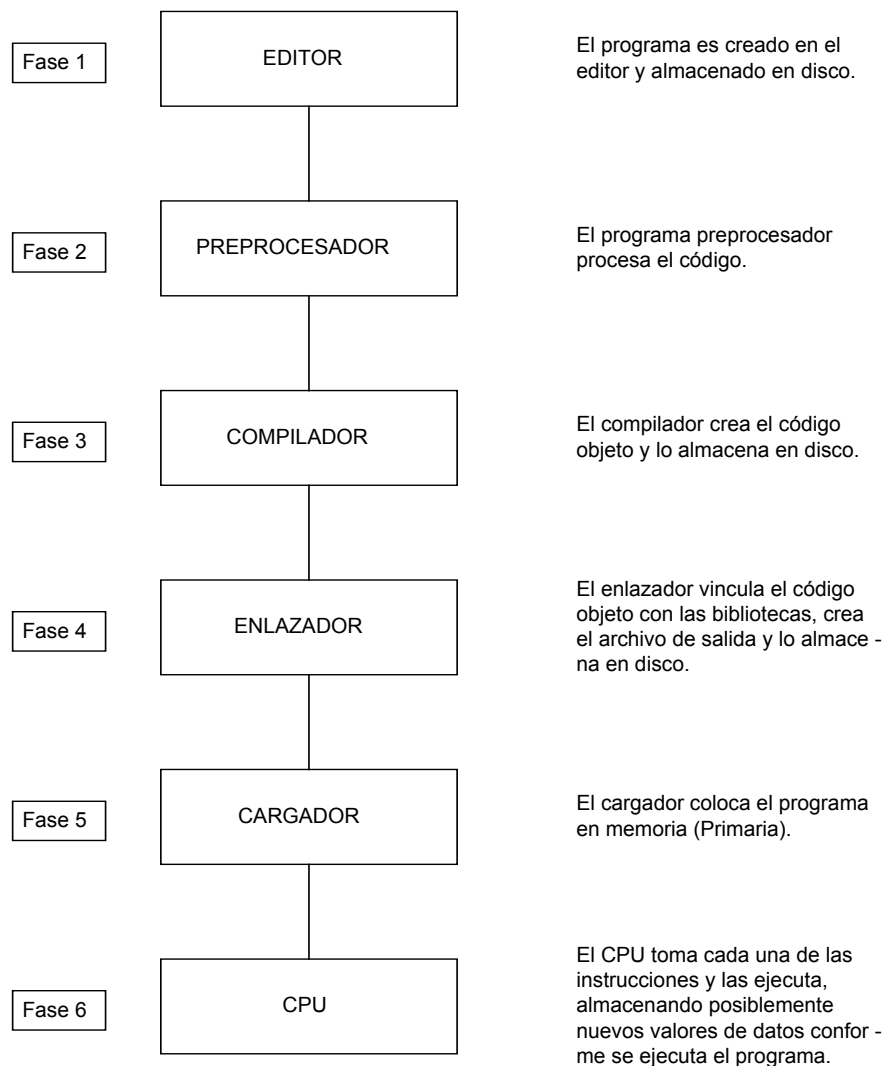
para el segmento. Luego, un Kbyte de memoria permite almacenar 256 vectores de interrupción. Algunos de los 256 vectores de interrupción son utilizados por el sistema, otros están libres para ser usados por los programas de usuario. El componente instala una rutina de interrupción en el canal de interrupción IRQ1, que es el reloj de sistema. Este timer genera una interrupción 18.2 veces por segundo. En la rutina de servicio de la interrupción, se incrementa una variable global. Cuando esta variable es igual a 18 se presenta en la pantalla.

### **5.3 Descripción del software utilizado para desarrollar la interfaz gráfica del reloj de cesio**

La implementación del software del sistema se efectuó bajo el lenguaje de programación Delphi 5, desarrollado por Borland. El entorno de desarrollo de Delphi es simple, flexible y potente al mismo tiempo; contando con un gran número de componentes prefabricados que simplifican de forma notable la creación de cualquier aplicación. El lenguaje base de Delphi, como se ha dicho antes es Object Pascal, Una característica que diferencia a Delphi de otros lenguajes, es el hecho de tratarse de un compilador altamente optimizado, que genera código directamente ejecutable, no pseudo-código que debe ser interpretado posteriormente en tiempo de ejecución. Esto tiene básicamente dos consecuencias: el ejecutable es más rápido, al no tener que ser interpretado en tiempo de ejecución; y el tamaño final total de la aplicación suele ser inferior, ya que no es necesaria la distribución adicional del programa que interpreta y ejecuta el código.

El lenguaje trabaja bajo un entorno típico de desarrollo, ejecutado a través de seis fases claramente marcadas (ver figura 5.4). Suponiéndose que se ha editado un programa se procede a dar el comando *compilar* el programa; el compilador traduce el programa Delphi a código de lenguaje de máquina (también conocido como código objeto), luego un programa *preprocesador* se ejecuta de forma automática antes de la fase de traducción, éste preprocesador obedece a comandos especiales llamados directrices o directivas de preprocesador, que indican que antes de su compilación se deben ejecutar ciertas manipulaciones sobre el programa. Estas manipulaciones por

lo general consisten en la inclusión de otros archivos en el archivo a compilar y en el reemplazo de símbolos especiales con texto de programa.



POWER-POINT 2000

**Figura 5.4** Entorno típico de Delphi 5i en la realización y ejecución de programas.

El preprocesador es invocado de manera automática por el compilador, antes que el programa sea convertido a lenguaje de máquina. La cuarta fase se conoce como enlace; los programas en Delphi 5 por lo general contienen referencias a funciones definidas en algún otro lugar como en bibliotecas estándar o en bibliotecas creadas para un proyecto en particular.

Entonces, el código objeto producido por el compilador, contendrá zonas vacías debido a estas partes faltantes. Un *Enlazador* vinculará el código objeto con el código de las funciones faltantes para producir una imagen ejecutable. La quinta fase se llama cargar; antes de que un programa pueda ser ejecutado, el mismo debe ser colocado en memoria, esto se lleva a cabo por el *cargador*, que toma la imagen ejecutable del disco y la transfiere a memoria. Por último, la computadora bajo el control de su CPU ejecuta el programa.

Una vez efectuados los procedimientos que hacen operar de forma eficiente las posibles acciones que conciernen a un reloj (direccionamiento y señal de un pulso por segundo), se introduce la interrupción que sustituyó a la interrupción original que controla los procesos de “timing” de la PC, esto con el efecto de buscar una mayor resolución en la toma de tiempos entre cada evento suscitado.

Las consideraciones de software en cuanto a la programación del PIC, fueron las siguientes:

1. Cargar el Registro de Control de Modo.
2. Escribir el valor de cuenta.
3. Si el contador 0 es modificado, se debe guardar el valor original de la INT 8 para que este sea llamado (actualización o refrescamiento) cada 18.2 veces por segundo. Cuando el manejador de la INT 8 no es llamado, no se debe enviar el fin de interrupción (EOI) al PIC (8259), hasta que se refresque el manejador original de la INT 8.

Este procedimiento se realiza en el programa cuando se carga la nueva interrupción. Dentro de las consideraciones es necesario tener en cuenta la relación que se establece entre el valor con que se carga el Timer (indica el valor hasta donde llegará de la nueva cuenta; por ejemplo desde 00H hasta FFH, cuenta que equivale a la cuenta normal) y la cantidad de veces que se tiene que dividir este valor, para obtener un avance de tiempo normal del reloj.

En lo que respecta a la interfaz gráfica desarrollada en Delphi 5, esta consta de 6 unidades, mismas que se describen a continuación, y cuyos procedimientos más relevantes se muestran en el apéndice 3:

### **5.3.1 Unidad Interfaz.dcu:**

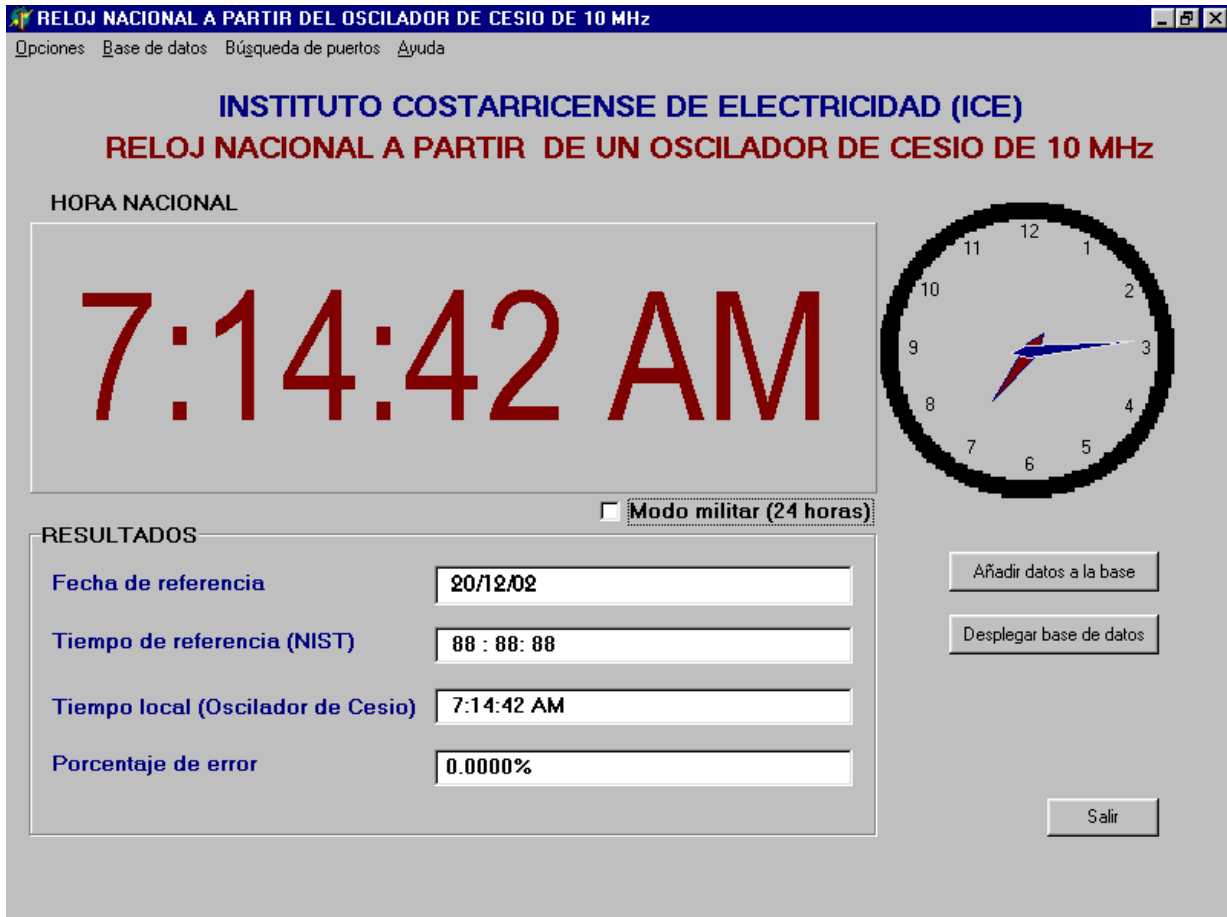
Es la unidad encargada de inicializar el programa de la interfaz gráfica. Como se explicó en el programa realizado en MPLAB, al programar el PIC, se modifica el oscilador interno de la PC, lo cual hace que la interrupción que genera la hora sea modificada también. De esta forma la computadora verá como su nuevo sistema de tiempo al tiempo generado desde el bus ISA, por el microcontrolador. La unidad Interfaz.dcu, genera toda la interfaz gráfica para el despliegue de este tiempo. La hora se desplegará sin necesidad de generar un código fuente muy extenso al invocar dentro del procedimiento TInterfaz\_usuario.Timer1Timer, la función TimetoStr(time). En este caso, la variable time, es el tiempo proveniente de la división del oscilador de cesio por un factor de 10, es decir, el tiempo que se genera desde el bus ISA. La función TimetoStr, convierte el parámetro de tiempo del bus ISA en un string, o cadena de caracteres de cualquier longitud.

En consecuencia, esto permite que se pueda visualizar la hora como se observa en la figura 5.5. Asimismo, la interfaz desarrollada permite que el usuario observe la fecha correspondiente. Es importante destacar, que la hora se puede desplegar en dos formatos según lo desee el usuario. Estos formatos son el militar y el tradicional.

Otra característica del programa es que se ha dejado prevista la conexión de una señal proveniente del Equipo del NIST, la cual dará el tiempo correcto, o tiempo de referencia; al igual que la fecha. Como esta señal no fue conectada al sistema en pantalla se puede observar en la figura 5.5 que la casilla denominada “Tiempo de referencia (NIST)”, tiene un valor indefinido, al igual que la casilla “Porcentaje de



error”, la cual se dejó prevista para que en futuras modificaciones a la interfaz, se pueda visualizar el porcentaje de error entre los dos sistemas.



DELPHI 5

**Figura 5.5** Interfaz gráfica del reloj patrón a partir del oscilador de cesio de 10MHz.

La interfaz gráfica del sistema, tiene un componente que despliega la hora en forma de reloj analógico. Este componente pertenece a la compañía Adrock, el cual ha requerido de un código de programación previo, lo que produce un retardo. Asimismo, la interfaz contiene un menú de inicio el cual desplegará la información referente a la base de datos, el estado de los puertos y una opción de ayuda.

Si el usuario así lo desea, puede acceder a la base de datos mediante una botón de acceso rápido dentro de la misma interfaz denominado “Desplegar base de

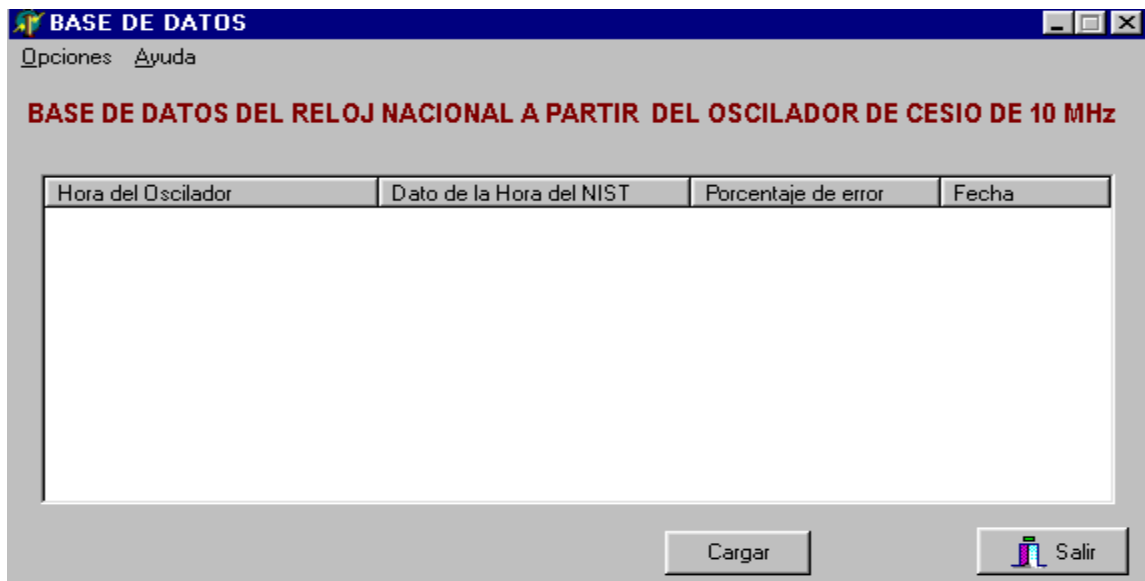
datos”. Si el usuario lo desea puede incluir algún tiempo fuera del intervalo definido por él para la toma de datos mediante la tecla “Añadir datos a la base”.

### **5.3.2 Unidad Basedatos.dcu:**

Es la unidad que despliega la base de datos del programa. La base de datos será un archivo de texto, el cual el programa va actualizando cada cierto tiempo. Este tiempo se encuentra en un intervalo de 60 a 3600 segundos. Es importante señalar que si el programa está corriendo por primera vez, el intervalo para la toma y almacenamiento de datos por defecto, será de 60 segundos. Lo anterior como medida de protección del sistema.

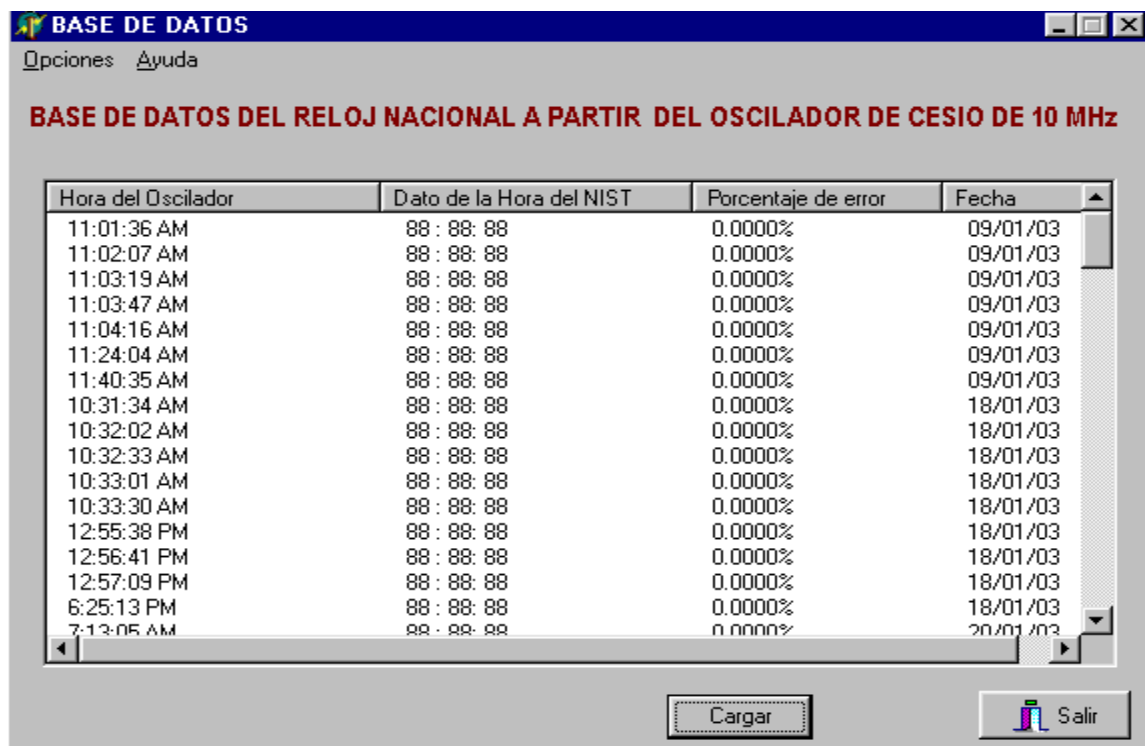
El archivo de texto estará dentro de la carpeta de instalación del programa y será un archivo de sólo lectura. Asimismo, dentro de esta carpeta este archivo posee la característica de que no puede ser eliminado de la misma. El archivo de texto se está actualizando cada 60 segundos (cuando el programa se corra por primera vez), o en el intervalo de tiempo establecido por el usuario.

Una vez que el usuario dentro de la interfaz gráfica quiera que se despliegue la base de datos solo deberá presionar en el menú o en la tecla de acceso rápida la opción correspondiente, abriendo una ventana aparte de la interfaz que se observa en la figura 5.6. La ventana se abre y se puede observar una serie de columnas para desplegar la información desde el archivo. Para acceder a la base de datos se debe presionar la tecla “cargar”, la cual abrirá el archivo y lo desplegará en las casillas correspondientes, como se muestra en la figura 5.7. Es importante señalar que esta información se toma de la interfaz gráfica, en las casillas de resultados.



DELPHI 5

Figura 5.6 Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Ventana de inicio).



DELPHI 5

Figura 5.7 Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Después de presionar la opción "Cargar").

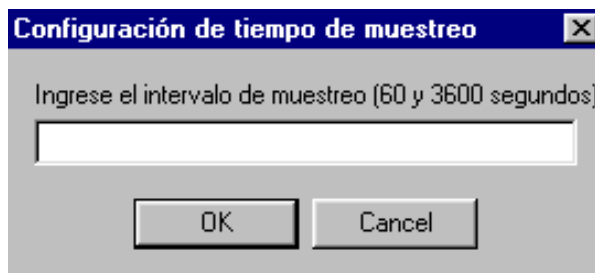
Estos datos se almacenan en el archivo basedatos.txt y se despliegan en la ventana dentro de la interfaz mediante un componente de Delphi 5 llamado TListView. Los datos se almacenan dentro del archivo de texto uno por línea (Ver apéndice A3, Interfaz.dcu); es decir, se almacena en la primera línea el tiempo del oscilador, en la segunda, el tiempo de referencia, en la tercera el porcentaje de error y finalmente la fecha de referencia en la cuarta. El ciclo se repite indefinidamente hasta que la aplicación finalice.

Una vez presionada la tecla “Cargar”, el TListView va a permitir visualizar los datos almacenados en el documento de texto, en forma de una lista, que es la que se visualiza en la figura 5.7. En su forma básica, un TListView almacena una colección de objetos denominados en Delphi “TListItem”. Cada TListItem tiene asociado un título o propiedad llamada Caption. Esta propiedad dentro del programa para la base de datos, se iguala dentro del código a la propiedad Caption de los objetos dentro de la interfaz gráfica. Por ejemplo, la propiedad Caption del objeto llamado en la interfaz gráfica “Tiempo local (Oscilador de Cesio)”, será igual a la propiedad Caption del primer TListItem del TListView (Ver apéndice A3, Interfaz.dcu), lo que provocará que al desplegarse los datos estos TListItem se desplieguen también.

En esta unidad al apretarse la opción “Cargar”, se va a invocar al procedimiento llamado Tbase.Button2Click (Ver apéndice A3, Basedatos.dcu), el cual se va a encargar de abrir el archivo de texto basedatos.txt de limpiar el TListView (para actualizarlo, en caso de que ya se haya abierto el archivo) y de ir añadiendo al TListView la información correspondiente a los cuatro datos obtenidos de la interfaz gráfica.

El programa le ofrece al usuario la posibilidad de configurar el tiempo de toma y almacenamiento de los datos (intervalo entre 60 y 3600 segundos), el cual se puede acceder dentro de esta ventana desde su menú, al seleccionar la opción “Configurar el tiempo de toma de datos”. De esta forma aparecerá una ventana de diálogo similar a la mostrada en la figura 5.8. Una vez que se ingrese el intervalo correcto el contador un contador para la toma de datos programado en la unidad

interfaz.dcu tendrá el nuevo valor escogido, produciendo un nuevo conteo y un nuevo intervalo de datos, mediante el procedimiento TInterfaz\_usuario.Timer1Timer (Ver apéndice A3, Interfaz.dcu). Si el intervalo que se ingresa no es el correcto el programa genera un aviso y vuelve a pedir un tiempo de toma de datos dentro del intervalo establecido para la programación. Es importante destacar que este intervalo se determinó de acuerdo a las exigencias del ICE. Si el usuario presiona la tecla Esc, cancela la opción de configuración y el tiempo de toma de datos será el tiempo por defecto, o 60 segundos.



DELPHI 5

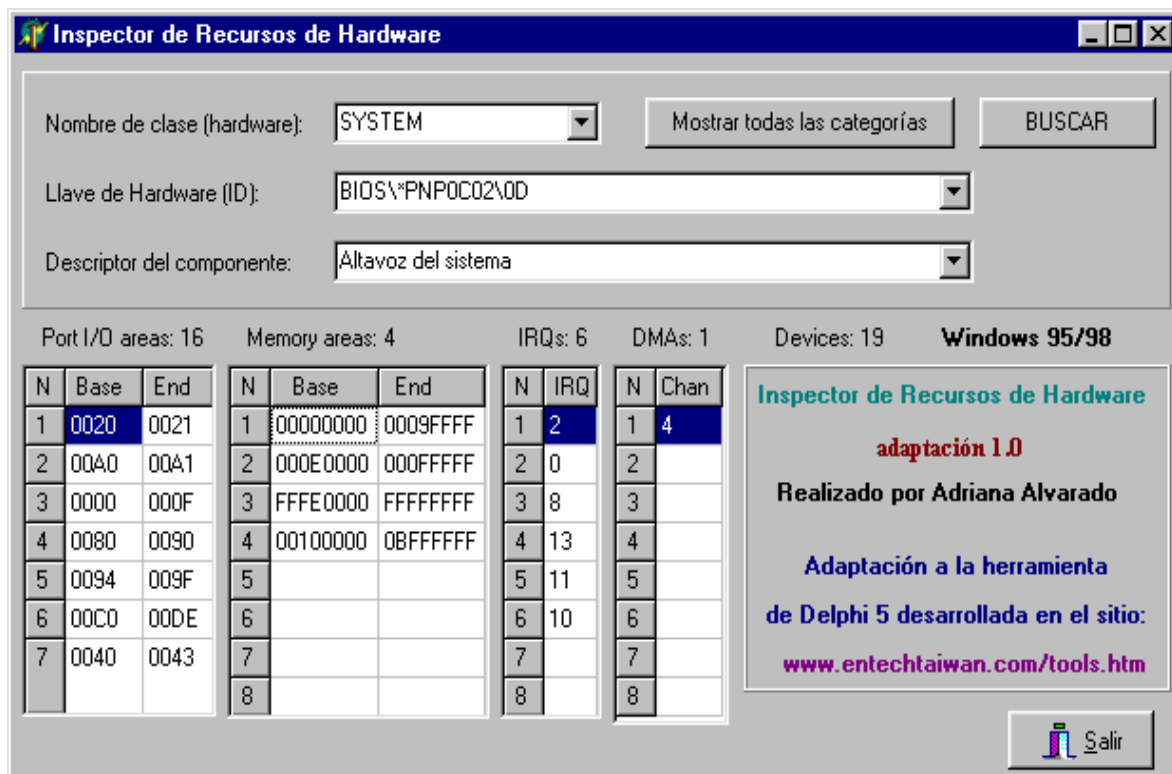
**Figura 5.8** Opción configuración de tiempo de muestreo del menú de la base de datos.

### 5.3.3 Unidad Main.dcu:

Es la unidad encargada de ayudar al usuario en el caso de que el sistema no este funcionando correctamente, debido a que le permite al usuario realizar un diagnóstico de todos los dispositivos incluidos en la computadora donde se está ejecutando el programa. Esta ventana es una adaptación a una herramienta desarrollada en el sitio [www.entechtaiwan.com/tools.htm](http://www.entechtaiwan.com/tools.htm), misma que se adaptó para el sistema Windows 95 o 98 y que se muestra en la figura 5.9.

Cuando se presiona la opción “Mostrar todas las categorías”, la unidad va a invocar a otra unidad llamada Vic\_Res (Ver apéndice A3, Vic\_Ress.dcu y Main.dcu), la cual va a realizar una búsqueda a través del sistema operativo de windows, de los recursos de hardware disponibles dentro de la PC. En consecuencia se van a desplegar una serie de clases de hardware dentro de la casilla llamada “Nombre de

clase (Hardware)”, que serán todos los recursos disponibles. Si se busca dentro de esta casilla la opción “SYSTEM”, y luego se presiona la tecla “Buscar” deberá aparecer en la casilla “Descriptor de Componente”, el recurso del bus ISA hallado en la PC. En caso contrario, el funcionamiento del sistema no se está dando a nivel de hardware. En consecuencia, Esta unidad servirá únicamente de referencia al usuario sobre el comportamiento del reloj.



DELPHI 5

**Figura 5.9** Inspector de recursos de hardware del reloj nacional a partir del oscilador de cesio de 10 MHz.

### 5.3.4 Unidad ClaveSalida.dcu:

Es la unidad encargada de generar un dialogo cuando el usuario desee salir de la aplicación. En el momento en el cual el usuario, seleccione ya sea del menú principal en la interfaz gráfica, o de la tecla de acceso rápido la opción “Salir”, se invocará un procedimiento llamado TInterfaz\_usuario.SalirClick (Ver apéndice A3,

Interfaz.dcu), el cual abrirá una pequeña ventana que se muestra en la figura 5.10. Esta ventana llamará a una función dentro de la unidad ClaveSalida.dcu llamada password la cual hará una comparación entre el dato ingresado por el usuario con el especificado para funcionar como password. Si el dato es correcto, el programa finalizará su ejecución; en caso contrario mantendrá la ventana de clave de salida abierta. Si el usuario presiona la tecla Esc del teclado, la ventana de clave de salida se cerrará y el programa continuará ejecutándose.



DELPHI 5

**Figura 5.10** Interfaz para la clave de acceso de salida del reloj nacional a partir del oscilador de cesio de 10 MHz.

### **5.3.5 Otras unidades del programa (Acerca.dcu y Ayuda):**

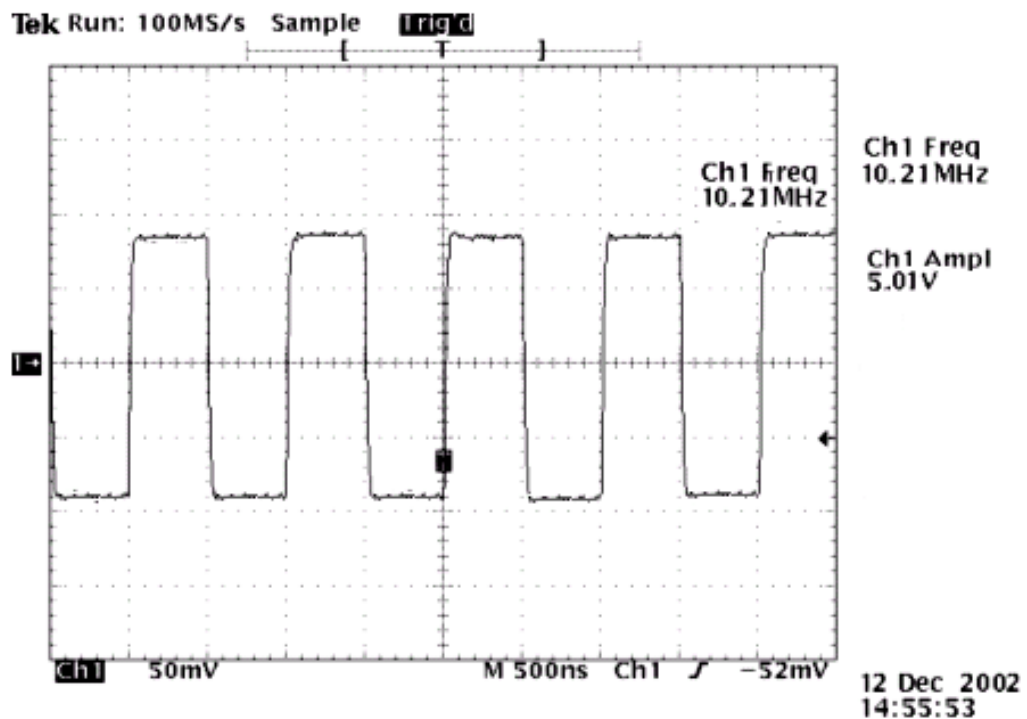
La unidad Acerca.dcu, es la unidad encargada de generar un diálogo el cual despliega la información acerca de la fecha de elaboración del programa, entidad para la cual se elaboró y demás créditos. La ayuda se despliega a partir de que el usuario selecciona la opción del menú Ayuda, la cual abre una ventana donde se desplegará un archivo con los temas más relevantes de la interfaz.

## **CAPITULO 6** **RESULTADOS EXPERIMENTALES**

---

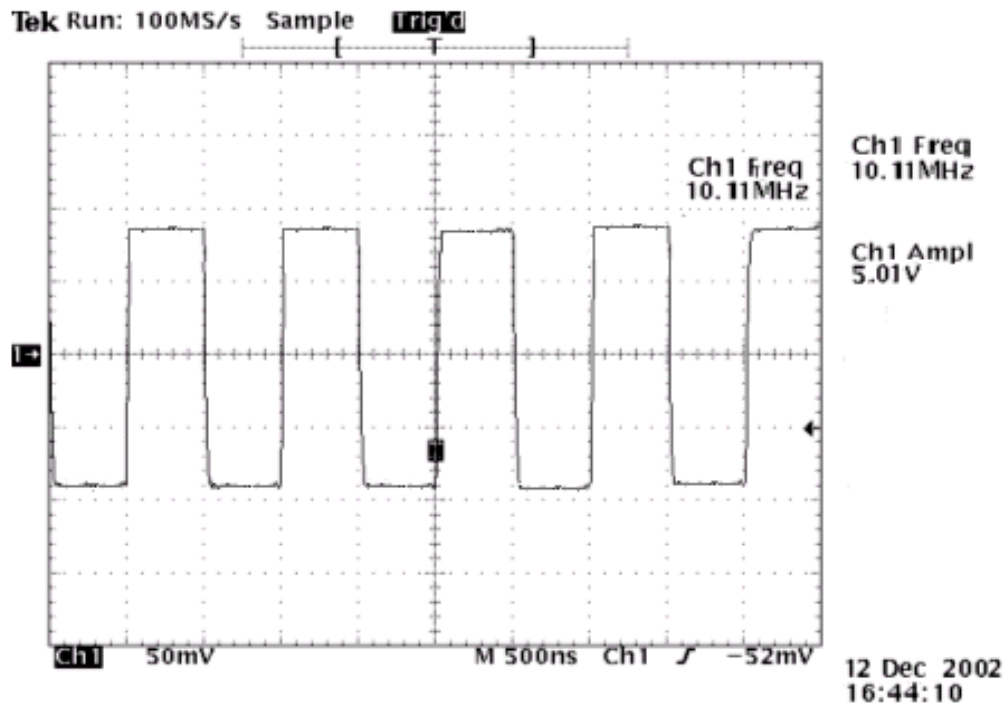
En este capítulo se muestran los resultados más representativos a nivel experimental, tanto de hardware como de software. Los oscilogramas mostrados en las figuras 6.1 y 6.2, muestran el efecto sobre el ruido que producen los diodos Zener conectados a las entradas de los BNC. Los diodos Zener no sólo limitan los voltajes y las corrientes mayores a las permitidas, sino que hacen que señales indeseables de ruido no sean introducidas en la tarjeta. Los diodos Zener en este caso, actúan como un filtro mediante el cual, las señales de ruido (pequeñas) van a ser disipadas por el par de diodos conectados en las terminales del BNC.

### **6.1 Oscilogramas del sistema**



**Figura 6.1** Oscilograma a partir del oscilador de cesio de 10 MHz. (Sin diodos Zener)





**Figura 6.2** Oscilograma a partir del oscilador de cesio de 10 MHz. (Con diodos Zener)

Asimismo, en la figura 6.3 se muestra el oscilograma obtenido en la salida del conector BNC5 (Ver figura 4.6). La salida deseada fue de 1Hz por segundo, la cual se observa en el oscilograma mencionado anteriormente. Es importante señalar que este oscilograma muestra la señal obtenida del proceso de división de la señal proveniente del oscilador de cesio de 10MHz.

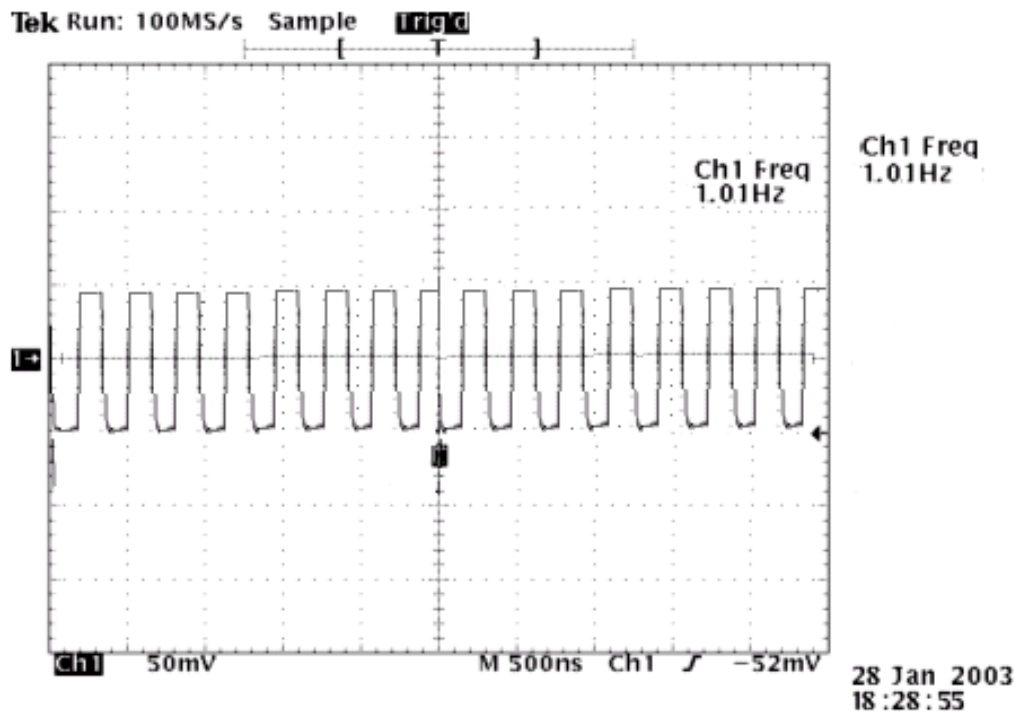


Figura 6.3 Oscilograma de la salida de 1 Hz.

## 6.2 Pruebas de exactitud

En la tabla 6.1 se muestran los resultados obtenidos en la salida de 1Hz por segundo; al igual que los porcentajes de error. Estos cálculos se efectuaron tomando como primicia, que el valor teórico de la frecuencia para la salida debió ser de 1Hz por segundo. En consecuencia, al obtenerse valores experimentales que oscilan entre 1.02 Hz y 1.04 Hz; el correspondiente porcentaje de error se tomó de confrontar los valores experimentales con el esperado teóricamente.

En la tabla 6.2 se muestran los resultados obtenidos con respecto a la diferencia en segundos del tiempo entre uno y otro almacenamiento de datos en la base. Como se observa en la tabla, y como se discutirá en el capítulo siguiente, existen tiempos de almacenamiento con diferencias entre 0 y 82 segundos, las cuales se deben en gran medida a la rapidez del equipo empleado en el desarrollo del sistema y a la operación de apertura de archivo (base de datos), actualización de datos y almacenamiento del nuevo archivo.

**Tabla 6.1** Porcentaje de error de valores obtenidos en la salida de 1 Hz.\*

| Valor obtenido (Hz) | % de error |
|---------------------|------------|
| 1.03                | 2.91262136 |
| 1.03                | 2.91262136 |
| 1.04                | 3.84615385 |
| 1.04                | 3.84615385 |
| 1.03                | 2.91262136 |
| 1.03                | 2.91262136 |
| 1.02                | 1.96078431 |
| 1.04                | 3.84615385 |
| 1.03                | 2.91262136 |
| 1.03                | 2.91262136 |
| 1.04                | 3.84615385 |
| 1.04                | 3.84615385 |
| 1.04                | 3.84615385 |
| 1.03                | 2.91262136 |
| 1.02                | 1.96078431 |
| 1.02                | 1.96078431 |
| 1.01                | 0.99009901 |
| 1.01                | 0.99009901 |

\*NOTA: Para los valores anteriores se utilizó un osciloscopio en la salida de 1 Hz.

**Tabla 6.2** Valores de tiempo para el almacenamiento de la base de datos .\*

| Tiempo medido | Fecha      | Margen de error (segundos) |
|---------------|------------|----------------------------|
| 03:59:17 p.m  | 20/01/2003 | 0                          |
| 04:58:59 p.m  | 20/01/2003 | 18                         |
| 05:58:23 p.m  | 20/01/2003 | 36                         |
| 06:59:34 p.m  | 20/01/2003 | 82                         |
| 07:58:07 p.m  | 20/01/2003 | 40                         |
| 08:59:42 p.m  | 20/01/2003 | 35                         |
| 09:58:36 p.m  | 20/01/2003 | 60                         |
| 10:59:21 p.m  | 20/01/2003 | 65                         |
| 11:58:07 p.m  | 20/01/2003 | 45                         |
| 12:59:47 p.m  | 20/01/2003 | 70                         |
| 01:59:07 a.m  | 21/01/2003 | 40                         |
| 02:59:12 a.m  | 21/01/2003 | 5                          |
| 03:59:34 a.m  | 21/01/2003 | 22                         |
| 04:59:45 a.m  | 21/01/2003 | 19                         |
| 05:58:16 a.m  | 21/01/2003 | 69                         |
| 06:59:26 a.m  | 21/01/2003 | 70                         |
| 07:58:55 a.m  | 21/01/2003 | 40                         |
| 08:59:32 a.m  | 21/01/2003 | 78                         |

\*NOTA: Muestra de valores anteriores a partir de un intervalo definido de 1 hora (3600 segundos), para el almacenamiento de valores en la base de datos.

## **CAPITULO 7** **ANÁLISIS DE RESULTADOS**

---

### **7.1 Explicación del diseño**

Para desarrollar el reloj patrón a partir del oscilador de cesio, una vez programado el PIC, se determinó la mejor forma de tomar el valor de los temporizadores internos de la PC, para introducirlos dentro de algún tipo de variable que fuera posible visualizar en pantalla y determinar gráficamente la resolución del dispositivo. El programa desarrollado para la programación del PIC funcionó y en un principio, se modificó el programa para el PIC en lo que respecta al registro de opciones debido a que la frecuencia se estaba dividiendo solo por factores múltiples de 2 y lo que se deseaba era que se dividiera por 10. Así que se procedió a dividir la señal de 10 MHz por un factor de 1 en el registro de opciones, para luego dividirla dentro de un divisor interno del PIC por un factor de 10. De esta forma se obtuvo el pulso por segundo.

Asimismo, en el programa se introdujo una variable que indicó el estado del bus ISA dentro de una interrupción, lo cual mejoró la detección de la señal del generador. Esta variable fue la encargada junto con otras dentro del programa del direccionamiento de los datos hacia el bus ISA.

El módulo de adquisición de datos se inicializará cuando el software reconoce el bus ISA. El microcontrolador se inicializa también y en la pantalla de despliegue de datos dentro de la aplicación desarrollada se comienza el conteo con el patrón de hora en formato normal (12:00:00, a.m o p.m). El Timer1 del microcontrolador es configurado para generar una interrupción cada segundo. El Timer1 rebasa el conteo cada segundo (se incrementa), por lo que la interrupción se genera y el microcontrolador deja el estado de reposo, lo que causa que los registros de tiempo programados dentro del algoritmo del controlador (horas, minutos y segundos), sean actualizados. Seguidamente, el microcontrolador estará en estado de reposo (SLEEP), hasta que reciba una nueva interrupción. Dentro del módulo de adquisición

de datos, se tienen tres modos de posicionar en forma manual el tiempo del reloj: el selector de unidades (Ver anexo A.2, señal S1) hace que se seleccione en forma manual cuales son las unidades que se desean modificar (horas o minutos), una señal encargada de incrementar el selector de unidades ( Ver anexo A.2, señal S2) y la señal CLR\_MIN (Ver anexo A.2, señal S3), la cual limpia el registro de minutos y segundos. Después de que la señal S1 o S2 se presionan, el usuario tiene 10 segundos para presionar la siguiente tecla. Si la segunda tecla no es presionada dentro de este intervalo de tiempo, la unidad (horas o minutos) retorna al sistema original y se sigue el conteo. En la interfaz gráfica cuando una de estas dos señales es presionada, el conteo sigue igual; y sólo cuando se presiona la segunda tecla se producirá un cambio en la interfaz gráfica.

El lenguaje de programación Delphi 5 brinda una función de acceso a estos “timers” mediante la utilización de una estructura de tiempo, en donde se pueden definir variables que son ligadas a la estructura y permiten la captura de las mediciones realizadas. Así, se pudo visualizar la cuenta común de un reloj, capturando el momento en que se genera el inicio de avance y la captura continua del tiempo actual, del temporizador según se definió en la estructura; se permite mediante la diferencia de los tiempos visualizar lo buscado. Se procedió a realizar todos los procedimientos que componen las funciones de un reloj digital, mediante un despliegue de análisis de tiempos por medio de la pantalla de una PC. La interfaz gráfica realizada con la PC funcionó, pero es importante señalar, que una vez que inicia se realizaba un conteo de forma ascendente; de manera que se detuviera hasta contar las 24 horas establecidas. En el momento en que este conteo llegaba a 24 horas, reiniciaba el mismo en 0 horas. La hora se puede desplegar en formato militar es decir si por ejemplo eran las 4:30 p.m, el programa marcaba las 16:30 p.m. Esta opción se desarrolló porque el programa del NIST despliega el tiempo de en los dos formatos antes mencionados.

Con respecto a la aplicación en Delphi 5, se pudo visualizar la cuenta común del reloj, capturando el momento en que se genera el inicio del conteo y la captura continua del tiempo por el temporizador (ver procedimiento TInterfaz\_usuario.

Timer1Timer, Interfaz.dcu; apéndice A3). El reloj o la señal de un pulso por segundo dividida se desplegó en pantalla, por medio de un contador ascendente. Para lograr esto se realizó todo un análisis previo de las características del bus ISA. Además, se creó un procedimiento en un programa aparte (Main.dcu), mediante el cual se pudiera indicar el estado de los puertos o buses, para poder determinar si la computadora estaba leyendo la señal del bus de estado).

La interfaz gráfica del sistema, tiene un componente que despliega la hora en forma de reloj analógico, el cual requirió de un código de programación previo, lo que produjo un retardo muy pequeño en el despliegue gráfico de los segundos. En consecuencia, si se activa la propiedad de mostrar los segundos que tiene el componente dentro del programa, se podrá observar este retardo. Este retardo no es significativo, pues solo se produce en el caso del reloj analógico. Lo anterior se da porque el programa tiene que tomar la hora del bus ISA y desplegarla por tres medios: las dos casillas de despliegue (donde el parámetro del tiempo se convierte en una cadena de string de cualquier magnitud) y a través del componente para generar el reloj analógico. Como se explicó, al ser este componente programado en forma previa, tiene un código que tomará el tiempo y lo convertirá en un parámetro de forma visual, empleando los números y las manecillas propias del reloj. Por lo tanto, al manejar mayores recursos visuales, el retardo es mayor.

En el capítulo 6 o de resultados, se observan los oscilogramas tomados a partir de la señal del oscilador de cesio de 10 MHz. En las figuras 6.1 y 6.2, se puede apreciar la influencia del ruido y la función de los diodos Zener al disminuir éste. Se puede ver que con los diodos Zener la señal de entrada se mejora, es decir, presenta menos influencia de ruido, debido a que los diodos Zener en este caso, actúan como un filtro mediante el cual, las señales de ruido (pequeñas) van a ser disipadas por el par de diodos conectados en las terminales del BNC. El oscilograma de la figura 6.3 muestra la señal obtenida del proceso de división de frecuencia. Según los requerimientos del sistema, la señal obtenida cumple con los objetivos, ya

que al ser una señal de 1 Hz, se le puede conectar a otro sistema de tiempo con características similares y ser comparadas entre sí.

La exactitud y la estabilidad del reloj patrón usando el Timer1 depende en forma directa del tipo de cristal u oscilador utilizado. En el caso de este sistema, al utilizarse un oscilador de cesio, cuya estabilidad es muy alta, produce que la señal de entrada por el microcontrolador no varíe en una forma brusca con el paso del tiempo. Adicionalmente, el uso de los diodos Zener filtra algunas señales de ruido, por lo que al efectuarse el proceso de división de frecuencia la señal resultante tiene una estabilidad alta. Lo anterior se comprueba al observar la tabla 6.1, ya que los valores obtenidos del conector BNC5 por donde sale la señal de 1Hz, se mantienen con pocas variaciones entre sí. Es importante destacar que los datos contenidos en esta tabla se tomaron en un periodo de 18 días, es decir, con una lectura del conector diaria; esto con el objetivo de analizar su estabilidad en intervalos prolongados de tiempo. Finalmente, se obtuvo los porcentajes de error pertenecientes a los valores obtenidos. Se observan que los porcentajes de error con respecto al valor deseado (1 Hz) se encuentran por debajo de un 4% como máximo (ver figura 6.1); lo cual demuestra que la salida experimental de 1 Hz no sufrió cambios significativos durante los 18 días de prueba. Esta situación se da por la estabilidad de la señal manejada, proveniente del oscilador de cesio.

En la tabla 6.2 se pueden observar los datos almacenados dentro del cuadro comparativo o base de datos de la aplicación. Para un intervalo definido por el usuario de 1 hora, se realizaron pruebas durante varios días, con el objetivo de analizar la precisión de los intervalos de almacenamiento dentro del programa. La tabla 6.2 presenta una serie de datos almacenados y los márgenes de error, los cuales son medidos en segundos. Los márgenes de error, se refieren a los segundos antes o después del intervalo seleccionado por el usuario, en los cuales el programa almacena los tiempos desplegados. Es así como en la tabla 6.2 para un intervalo de 3600 segundos, el margen más alto es de 82 segundos. Uno de los factores que pudo influir dentro de este margen fue el del tiempo de almacenamiento del dato

dentro de la base, debido a que el programa busca el archivo de texto destinado para almacenar el dato, se ubica en el último dato almacenado y actualiza la base guardando el nuevo dato. Si el archivo es sumamente grande (ya ha almacenado gran cantidad de datos en la base), en el momento en que se actualice el archivo durará mayor tiempo. Asimismo, si en el computador se están ejecutando otras aplicaciones en forma simultánea a la del reloj patrón, la toma de datos se verá afectada debido a que el computador no posee una memoria suficientemente grande para manejar con eficiencia el almacenamiento de estos datos; de ahí que el margen de error sea tan variable dentro de la tabla 6.3

Uno de los mayores problemas que se plantearon al diseñar la aplicación, fue el de acceder a una base de datos, en lo que se refiere al método para determinar su formato. Con respecto a la base de datos que se creó, se comprobó que funciona de acuerdo con los requerimientos, debido que toma lecturas en intervalos de una hora actualizando y guardando el archivo; además de desplegarlo si el usuario así lo requiere mediante la opción del menú llamada “*Base de datos*”. Esta base de datos, quedó guardada en un archivo de texto en la carpeta donde se instaló el programa, de forma tal que se puede acceder directamente de la carpeta. Este archivo como se explicó en el capítulo anterior, se actualiza al medir cada hora los datos.

En la base de datos se puede visualizar una columna donde se despliega el error. Este error es el resultado de la sustracción entre la hora generada por el oscilador de cesio y la hora generada por el equipo del NIST. Para determinar el error introducido por la computadora utilizada, se dejó previsto realizar en futuras versiones de este prototipo, comparaciones del sistema diseñado con el patrón primario del NIST, la desviación existente entre este patrón y la PC, representa el error introducido por la computadora. Este valor se debe restar a cada medición final, para corregir la desviación del “instrumento”.

Es importante señalar que la comparación entre el sistema diseñado con una señal proveniente del NIST no se realizó. No obstante el sistema desarrollado tanto a nivel de software como de hardware, se encuentra en condiciones para recibir la señal del NIST; y en futuras versiones realizar los procesos de comparación y



despliegue de porcentajes de error. Entre mayor sea el periodo de comparación, más exacto será el valor de la desviación obtenida.

Una vez obtenido el aumento en la precisión y la visualización en pantalla, se procedió a corregir el problema que se presentaba por las características del bus ISA y de la computadora en cuanto a la protección de sobrevoltajes y cortocircuitos. El diseño de la tarjeta de adquisición de datos fue orientado a brindar este tipo de protección. Se diseñó un circuito limitador de voltaje que se encarga de mantener el voltaje de alimentación de la tarjeta a un valor de  $5 V_{cd}$ , propio para el suministro de tecnología TTL, para tener una alimentación estable y que proteja la integridad de los “chips” utilizados.

## **7.2 Alcances y limitaciones**

Se logró cumplir con la estructura que fundamentó la realización del proyecto: un prototipo capaz de generar un patrón de tiempo a partir de un oscilador de cesio de 10MHz de una manera automática. El prototipo preliminar permite la realización de cambios que harán de éste un sistema más versátil en el futuro.

Se implementó el prototipo que brindará alta confiabilidad y accesibilidad, lo cual se demuestra en el desarrollo de una rutina dentro del sistema que revisa constantemente la lectura del bus ISA. Con la utilización de los procedimientos que se encargan de verificar el estado del sistema, se permitirá determinar de una forma más rápida y sencilla la posible causa de algún problema; evitando pérdidas de tiempo en la revisión del software o hardware. Se logró obtener una resolución mayor a la ofrecida por la PC; se obtuvo un sistema 256 veces más preciso al que se tenía en la situación inicial, aspecto que constituye uno de los puntos de mayor peso en cuanto a objetivos se refiere.

En relación con el hardware, el programa presenta una limitación de importancia, ya que las frecuencias intermedias generadas por el PIC no podrán ser cambiadas a menos que se sustraiga del circuito el microcontrolador y se vuelva a reprogramar con el nuevo intervalo. Lo anterior es una ventaja porque ofrece

protección y confiabilidad, ya que evita que cualquier persona pueda intervenir en el sistema. No obstante, presenta ciertas dificultades porque si se desea cambiar los rangos de frecuencias para determinada medición no se pueden efectuar a menos que se cuente con el código del programa, se entienda el código, se pueda estar en capacidad de modificarlo y finalmente, se tenga el programador adecuado para hacerlo.

Una limitación importante de este prototipo, es que no realiza la comparación de la señal de 1Hz con una señal proveniente del NIST. Lo anterior hace que haya un porcentaje de error de un 0% dentro del cuadro comparativo mostrado en la figura 5.7, dado que la señal de un 1Hz no está siendo comparada con ninguna otra. No obstante, el prototipo deja prevista esta opción para futuras modificaciones del mismo. Asimismo, es importante hacer referencia que se trabajó con una computadora cuyos recursos de hardware limitan el funcionamiento del prototipo desarrollado. Por ejemplo, la memoria de la computadora utilizada impide que se ejecuten muchas tareas al mismo tiempo. En consecuencia, es preferible que la aplicación de software del prototipo sea la única aplicación funcionando, esto con el objetivo de que no haya conflictos en el momento de manejar información específica del prototipo, como por ejemplo, el manejo del archivo de texto correspondiente a la base de datos diseñada.

Finalmente, el programa que se encarga del manejo estadístico de los datos se desarrolló dentro de la misma aplicación de Delphi5. Sin embargo, no se pudo implementar una base de datos que creara archivos en forma diaria, semanal o mensual, en forma individual dentro del disco de la computadora, pues para el registro físico de los datos se programó un tipo de base de datos que manejaba un solo archivo y no generaba archivos individuales según día semana o mes. Lo anterior porque crear bases por semana, o mes no se puede realizar por medio de archivos de texto en Delphi 5.

Debido a que en el Laboratorio Metrológico los intervalos de muestreo no son muy cortos se puede manejar los datos a ser desplegados en pantalla mediante un archivo de texto, similar al confeccionado en esta aplicación. Sin embargo, este

archivo irá creciendo conforme se vayan almacenando los diferentes datos en forma infinita, por lo que a largo plazo, si el departamento lo desea puede eliminar los archivos; o tendrá que modificar el código trabajando en bases especialmente diseñadas en Delphi para trabajar con un servidor.

## **CONCLUSIONES Y RECOMENDACIONES**

---

### **CONCLUSIONES**

1. La introducción de una variable que indique el estado del bus ISA dentro de una interrupción, hará más exacta (aproximadamente 256 veces la velocidad de la computadora) la detección de la señal del osciloscopio.
2. Las características de velocidad del bus ISA sumadas a las de ejecución interna de la PC, hacen del sistema desarrollado un medio más fiable y preciso en la toma de tiempos.
3. La ejecución de secuencias de verificación del sistema (pruebas de error), en sistemas de medición, facilitan la determinación previa a posibles errores que podrían generar algunas inconsistencias en la toma de mediciones.
4. Para procesos en donde se requiera de alta precisión, la utilización de procedimientos que involucren los temporizadores de la PC para la toma de mediciones, serán una solución viable y efectiva.
5. Al introducir un factor de corrección en la toma de tiempos, dentro de la ejecución de una interrupción, se hace posible mejorar la precisión ofrecida por la computadora, en una relación de  $n$  veces más por segundo; donde  $n$  representa la cantidad de muestras posibles entre cada TIC de reloj.
6. El uso de un cuadro comparativo o base de datos para el control de registros de información, constituye una poderosa herramienta que ayuda a mejorar los procesos de consulta y de manejo de memoria en una PC.

7. La forma en que se manipulan las interrupciones en el lenguaje Delphi 5 hace que sean manipulables también bajo el ambiente Windows.

8. Para la protección y seguridad del sistema desarrollado, es imprescindible incluir equipos de protección contra sobrevoltajes, tales como fuentes supresoras de picos de corriente, así como sistemas de alimentación preventiva (UPS). Se debe contar con un sistema eléctrico totalmente acoplado (utilización de tierras másicas, además de las eléctricas).

9. El ruido es un factor de mucha influencia en procesos de medición de tiempo, pues puede generar frecuencias indeseables que perjudiquen los resultados finales.

10. El registro de opciones que tienen los microcontroladores en forma interna, facilitan los procesos de división de frecuencia, debido a su velocidad y precisión.

11. La exactitud y la estabilidad del reloj patrón usando el microcontrolador PIC 16F877 depende en forma directa del tipo de cristal u oscilador utilizado.

## **RECOMENDACIONES**

1. Para la protección y seguridad del prototipo, será necesario contar con equipo de protección de sobrevoltajes, tales como fuentes supresoras de picos de corriente, así como sistemas de alimentación preventiva (UPS), con el fin de proteger de una mejor forma los dispositivos del módulo de adquisición de datos un ambiente abierto.

2. En lo que respecta a la elaboración de las bases de datos semanal y mensual en Delphi 5, se mencionan a continuación las posibles herramientas a utilizar:

- a) Mediante DBD: Delphi incorpora la utilidad llamada *Database Desktop*, DBD, la cual facilita la creación y manipulación de bases de datos de forma

interactiva, sin necesidad de crear ninguna ficha o unidad de programación, ni escribir una sola línea de código. Desde DBD se pueden crear consultas, manipular los datos de la base, copiar datos entre bases, etc.

- b) Mediante BDE: El soporte de bases de datos en Delphi 5 está basado en lo que se conoce habitualmente como BDE, que son las siglas de *Borland Database Engine* (Motor de bases de datos de Borland). Delphi usa el BDE como un traductor o una capa intermedia entre las aplicaciones y las bases de datos a las que se deseen acceder. El BDE, es común a una aplicación indistintamente del formato de bases de datos que esta utilice. De esta forma, la operación para actualizar un dato u obtener un registro de una tabla siempre será igual desde un programa.
- c) Mediante SQL: En la gestión de bases de datos existe un lenguaje prácticamente estándar, conocido como SQL; el cual puede ser usado para realizar cualquier tipo de consulta, extracción y modificación de los datos de una base. SQL, es un lenguaje estructurado de consulta; sin embargo sólo es posible utilizarlo con aquellas bases de datos que lo soportan, como los servidores SSQL Server.

3. Se puede generar un procedimiento que de un formato oficial (hh:mm:ss:ms) a la presentación en pantalla, ya que la presentación actual, muestra únicamente el transcurso del tiempo en segundos. Lo anterior, con el propósito de hacer más eficiente en lo que respecta al conteo de tiempo del sistema.

4. Existen formas alternativas de efectuar el proceso de división de la señal proveniente del oscilador de cesio de 10 MHz. Una de ellas es introducir la señal del oscilador de cesio por 10 divisores de década conectados en cascada. La señal resultante debe ser de 1Hz por segundo. Asimismo, en los procesos de división dados por cada uno de los divisores de década, se pueden tomar frecuencias intermedias en el orden de divisores de 10. Sin embargo, la cantidad de circuitería

requerida para el proceso de división de frecuencia puede hacer que existan retardos significativos en la señal final deseada de 1 Hz por segundo.

5. Es importante hacer referencia que se trabajó con una computadora cuyos recursos de hardware limitan el funcionamiento del prototipo desarrollado. Por ejemplo, la memoria de la computadora utilizada impide que se ejecuten muchas tareas al mismo tiempo. Asimismo, la aplicación de software del prototipo debe ser la única aplicación funcionando, esto con el objetivo de que no haya conflictos en el momento de manejar información específica del prototipo, como por ejemplo, el manejo del archivo de texto correspondiente a la base de datos diseñada. Por este motivo se recomienda modificar la computadora desde la cual se ejecute la parte de software desarrollada en el prototipo; o se mejore algunas de las características de hardware de la misma como por ejemplo la cantidad de memoria. Dado que la aplicación del prototipo debe trabajar con archivos de texto y está en la capacidad de abrir ventanas de trabajo en forma simultánea y dentro de la misma aplicación; se recomienda incrementar la cantidad de memoria con el objetivo de mejorar la velocidad de respuesta de la computadora con respecto a la aplicación desarrollada.

## **BIBLIOGRAFÍA**

---

### **- Libros consultados:**

1. Angulo Usategui, José María. Microcontroladores Pic: Diseño práctico de aplicaciones. Segunda parte: PIC 16F87X. Segunda edición. Editorial McGraw-Hill: Madrid, España.2000
2. Cantu, Marco. Mastering Delphi 4. Primera edición. Editorial SYBEX: San Francisco, Estados Unidos,1998.
3. Charte, Francisco. Programación con Delphi 5. Segunda edición. Ediciones Anaya Multimedia: Madrid, España, 1999.
4. Coughlin Robert. Amplificadores operacionales y circuitos integrados lineales. Quinta edición. Editorial Prentice Hall: México, 1999.
5. Diettel H. Cómo programar en C++. Tercera edición. Editorial Pierce: Alabama, Estados Unidos.1997
6. Hoffman G. Edward , “Instrumentos Básicos de Medición”; Grupo Impresa, México, D.F,1992; p.p 19-23,37-43.
7. Jurgens David, “Hardware Data and Specifications “; Programa de Software, Aplicación (Help PC), 1991.
8. Lombardi, Michael. NIST Time and Frequency Services. NIST special publication 432, 2002 Edition.
9. Lucchesi Domenico, “Metrotecnica. Tolerancias e Instrumentación”; Editorial Labor, Barcelona,1973; p.p 19-21,70-77.
10. Malvino, Albert Paul. Principios de electrónica. Quinta edición. Editorial McGraw-Hill: México, 1997.
11. Nelson, Victor. Análisis y diseño de circuitos lógicos digitales. Primera edición. Editorial Prentice-Hall Hispanoamericana S.A: México, 1996.

### **- Manuales consultados:**

12. Microchip. Manual para MPLAB. Cd’s del fabricante.2000
13. Microchip. Hoja de especificación del microcontrolador PIC 16F877. Cd’s del fabricante.2000



**-Direcciones en Internet consultadas:**

14. "Digital Interface Cable "  
[http:// www.mars-am.com/cable/mogami/mtc/mtc56.html](http://www.mars-am.com/cable/mogami/mtc/mtc56.html)
15. Página Oficial de Microchip. [http:// www.microchip.com](http://www.microchip.com)
16. Private Users System: Programador de PICs.  
[http:// www.lokekieras.es /personales/mgsanz/programa.html](http://www.lokekieras.es/personales/mgsanz/programa.html)
17. Microcontroladores: Información, Herramientas y Programador.  
<http://www.geocities.com/TheTopics/2174/micro.html>
18. Microchip Net Resources. <http://www.geocities.com/SiliconValley/Way/5807>
19. Mundo electrónico : Enlaces.  
[http:// www.geocities.com/CapeCanaveral/Campus/9468/mundo.html](http://www.geocities.com/CapeCanaveral/Campus/9468/mundo.html)
20. Inspector de recursos de hardware. <http://www.entechtaiwan.com/tools.htm>

## **APÉNDICES**

## APÉNDICE A.1 Glosario de términos y lista de abreviaturas

- **Calibración:** Conjunto de operaciones que establecidos, bajo condiciones específicas, dan la relación entre los valores de magnitudes indicados por el instrumento o sistema de medición, o los valores representados por una medida materializada o un material de referencia y los correspondientes valores reportados en los patrones.

- **Confiabilidad:** Es aquella condición en la cual los resultados reales son iguales a los resultados deseados o previstos.

- **Error de medida:** Desviación entre la indicación dada por los instrumentos de medición y la verdadera magnitud de lo que se mide.

- **Exactitud:** Es el grado de conformidad con una norma establecida. Se puede considerar como una comparación entre los resultados deseados y los reales.

- **Laboratorio Metrológico (ICE):** Laboratorio encargado de custodiar los patrones metrológicos y la realización de las calibraciones de instrumentos y equipos de medición con trazabilidad NIST. Además se encarga de la caracterización de instrumentos y equipos de medición; y el asesoramiento en mediciones y en la selección de equipos de medición.

- **NIST:** Siglas del National Institute of Standards and Technologies. Esta institución ofrece servicios de calibración de equipo por electromagnetismo, calibración mecánica, por radiación óptica o termodinámica; y por tiempo y frecuencia. Este último tipo de calibración es el que esta institución proporciona al Laboratorio de Metrología del ICE.

- **Norma:** Es un valor establecido y conocido, que se utiliza para medir una cantidad desconocida.
  
- **Precisión:** Es la fidelidad del proceso de medición con relación a su repetibilidad.
  
- **Punto de referencia de una medición:** Base desde la cual se toma la medición, o punto donde empieza una medición.
  
- **Trazabilidad de una medida:** Propiedad del resultado de una medición o el valor de un patrón, por el cual puede ser relacionado con los patrones de referencia, usualmente patrones nacionales o internacionales, a través de una cadena ininterrumpida de comparaciones.
  
- **Tolerancia:** Es la cantidad total de variación permitida para una medida, con respecto a un valor nominal establecido. También conocida como incertidumbre, representativa de los límites máximo o mínimo dentro de los cuales puede fluctuar una medida para considerarse aceptable.

## APÉNDICE A.2 Circuito preliminar para la tarjeta de adquisición de datos

**Tabla A.2.1** Listado de las principales señales para la interfaz con el bus ISA

| SEÑAL                                  | DESCRIPCION   |
|--|---|
| A0-A19 (pines A31 a A12)               | Bits de dirección 0-19, permiten direccionar 1Mb de memoria y 64K de puertos de e/s.  |
| AEN (A20)                              | Address Enable; Cuando está activa el controlador DMA posee el control de las líneas de dirección y del BUS de datos, conforme se indique en MEMR/MEMW. Cuando está inactiva la CPU tiene el control de estas líneas. |
| ALE (B28)                              | Address Latch Enable (salida); se emplea para que la CPU esté aislada de las líneas de dirección (triestado). Es forzado activado durante los ciclos DMA.   |
| CLK                                    | Señal de reloj del sistema (actual velocidad del BUS).  |
| D0-D7 (A9 a A2)                        | Bits de datos 0-7 para e/s a memoria o puertos de e/s.  |
| RBO- RB3                               | Señales de transporte de datos del PIC a la tarjeta interfaz ISA  |
| RA0- RA2                               | Señales de control hacia el bus ISA (son los pines del PIC16F877 encargados de generar las interrupciones para lectura, junto con la interrupción del bus ISA hacia el computador)                                    |
| Señal del oscilador de cesio de 10 MHz | Señal que proviene del oscilador de cesio de 10 MHz y entra por el conector J1 (que lleva el nombre)  |
| Señal de 1 Hz                          | Señal que proviene del proceso de división de frecuencia (o de la señal del oscilador de cesio de 10 MHz. Es una señal saliente por el conector J5 (que lleva la etiqueta señal de 1 Hz)                              |



ICE

**RELOJ PATRÓN A PARTIR  
OSCILADOR DE CESIO  
10 MHz**

**MANUAL DEL USUARIO**

**GUÍA RÁPIDA**

**ELABORADO POR  
ADRIANA ALVARADO SALAZAR.**

**ENERO, 2003.**

## INDICE GENERAL

| CONTENIDO  | PAGINA |
|--|--------|
| Capítulo 1: Generalidades del programa .....                         | 1      |
| 1.1 Descripción general del programa .....                           | 1      |
| 1.2 Menú principal .....   | 2      |
| 1.3 Teclas de acceso rápido .....                                    | 5      |
| Capítulo 2: Base de datos .....                                      | 6      |
| 2.1 Tecla de acceso rápida “Cargar” .....                            | 7      |
| 2.2 Opción de configuración de Tiempo de muestreo .....              | 8      |
| 2.3 Ayuda .....  | 8      |
| Capítulo 3: Búsqueda de puertos: Estado de los puertos de la PC .... | 8      |
| 3.1 Aspectos generales del inspector de recursos .....               | 8      |
| Capítulo 4: Ayuda y mal funcionamiento del sistema .....             | 9      |
| 4.1 Tecla de ayuda .....   | 9      |
| 4.2 Mal funcionamiento del sistema .....                             | 9      |



## **Capítulo 1: Generalidades del programa**

### **1.1 Descripción general del programa**

La Interfaz del sistema, genera toda los requerimientos para el despliegue en tiempo real. En consecuencia, esto permite que se pueda visualizar la hora como se observa en la figura A.4.1. Asimismo, la interfaz desarrollada permite que el usuario observe la fecha correspondiente. Es importante destacar, que la hora se puede desplegar en dos formatos según lo desee el usuario. Estos formatos son el militar y el tradicional. Para acceder a cualquiera de los dos formatos anteriormente mencionados, debe marcar con el mouse la opción modo militar.

Otra característica del programa es que se ha dejado prevista la conexión de una señal proveniente del Equipo del NIST, la cual dará el tiempo correcto, o tiempo de referencia; al igual que la fecha. Como esta señal no fue conectada al sistema en pantalla se puede observar en la figura 1 que la casilla denominada “Tiempo de referencia (NIST)”, tiene un valor indefinido (88:88:88), al igual que la casilla “Porcentaje de error”, la cual se dejó prevista para que en futuras modificaciones a la interfaz, se pueda visualizar el porcentaje de error entre los dos sistemas.

La interfaz gráfica del sistema, tiene un componente que despliega la hora en forma de reloj analógico. Este componente pertenece a la compañía Adrock, el cual ha requerido de un código de programación previo, lo que produce un retardo. Asimismo, la interfaz contiene un menú de inicio el cual desplegará la información referente a la base de datos, el estado de los puertos y una opción de ayuda.

La interfaz gráfica del sistema, tiene un componente que despliega la hora en forma de reloj analógico. Este componente pertenece a la compañía Adrock, el cual ha requerido de un código de programación previo, lo que produce un retardo. Asimismo, la interfaz contiene un menú de inicio el cual desplegará la información referente a la base de datos, el estado de los puertos y una opción de ayuda.



Figura A.4.1 Interfaz gráfica del reloj patrón a partir del oscilador de cesio de 10MHz.

## 1.2 Menú principal

El usuario cuenta con un menú principal, el cual consta de 4 opciones, mismas que se detallan a continuación:

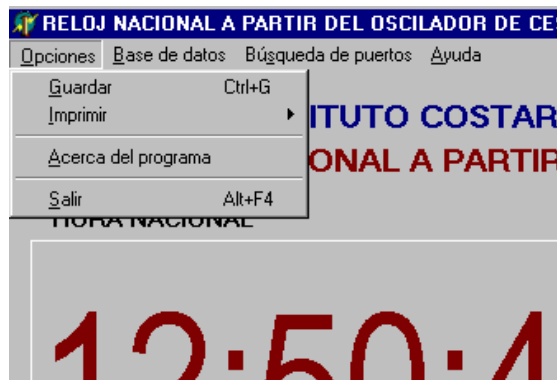


Figura A.4.2 Items de selección en el menú principal (Opciones)

### 1.2.1 Opciones:

El menú de opciones posee cuatro opciones : Guardar, Imprimir, Acerca del programa y salir.

**a. Guardar:** El usuario puede salvar a disco de 3 <sup>1</sup>/<sub>2</sub>, la base de datos hasta ese momento realizada. Al seleccionar esta opción aparecerá un diálogo en el cual el usuario puede guardar la base de datos en modo texto.

**b. Imprimir:** El usuario puede imprimir la base de datos hasta ese momento realizada. Al seleccionar esta opción aparecerá un diálogo en el cual el usuario puede imprimir la base de datos en modo texto, así como configurar el modo de impresión o la impresora.

**c. Acerca del programa:** Al seleccionar esta opción se generará un diálogo el cual despliega la información acerca de la fecha de elaboración del programa, entidad para la cual se elaboró y demás créditos.

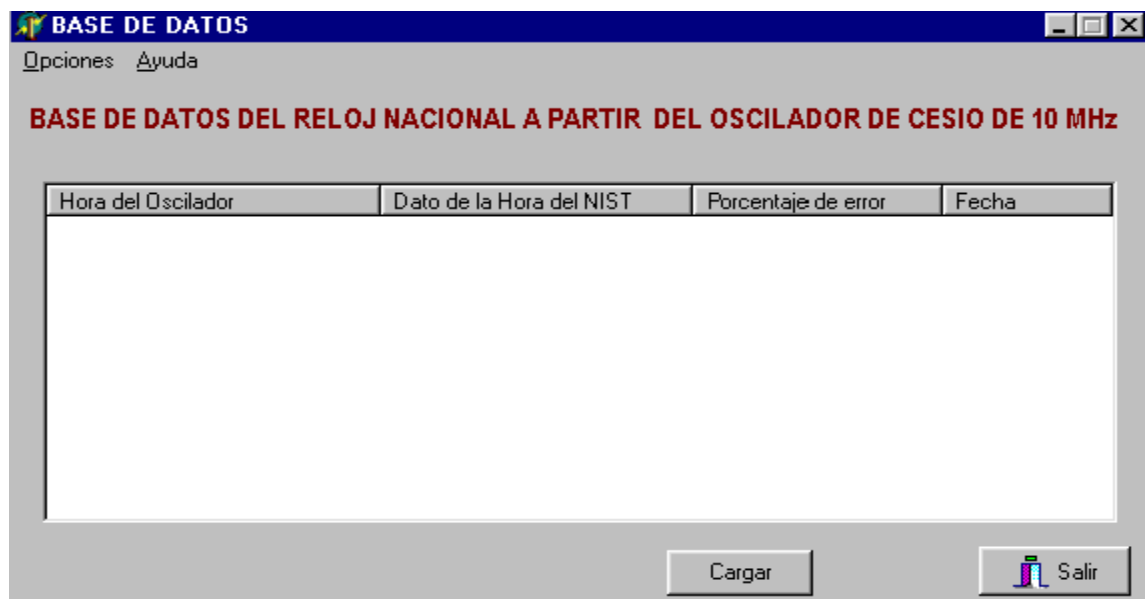
**d. Salir:** En el momento en el cual el usuario, seleccione esta opción, se generará un dialogo cuando desee salir de la aplicación. Esta opción abrirá una pequeña ventana que se muestra en la figura A.4.3, para que se ingrese la clave de salida. Una vez ingresada la contraseña, se hará una comparación entre el dato ingresado por el usuario con el especificado para funcionar como password. Si el dato es correcto, el programa finalizará su ejecución; en caso contrario mantendrá la ventana de clave de salida abierta. Si el usuario presiona la tecla Esc del teclado, la ventana de clave de salida se cerrará y el programa continuará ejecutándose.



**Figura A.4.3** Interfaz para la clave de acceso de salida del reloj nacional a partir del oscilador de cesio de 10 MHz.

### 1.2.2 Base de datos:

Al seleccionar esta opción, se abrirá una ventana igual a la mostrada en la figura A.4.4. En el capítulo “Base de datos” se discutirá su funcionamiento.



**Figura A.4.4** Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Ventana de inicio).

### 1.2.3 Búsqueda de puertos:

Al seleccionar esta opción, se abrirá una ventana igual a la mostrada en la figura A.4.5. En el capítulo “Búsqueda de puertos: Estado de puertos de la PC” se discutirá su funcionamiento.

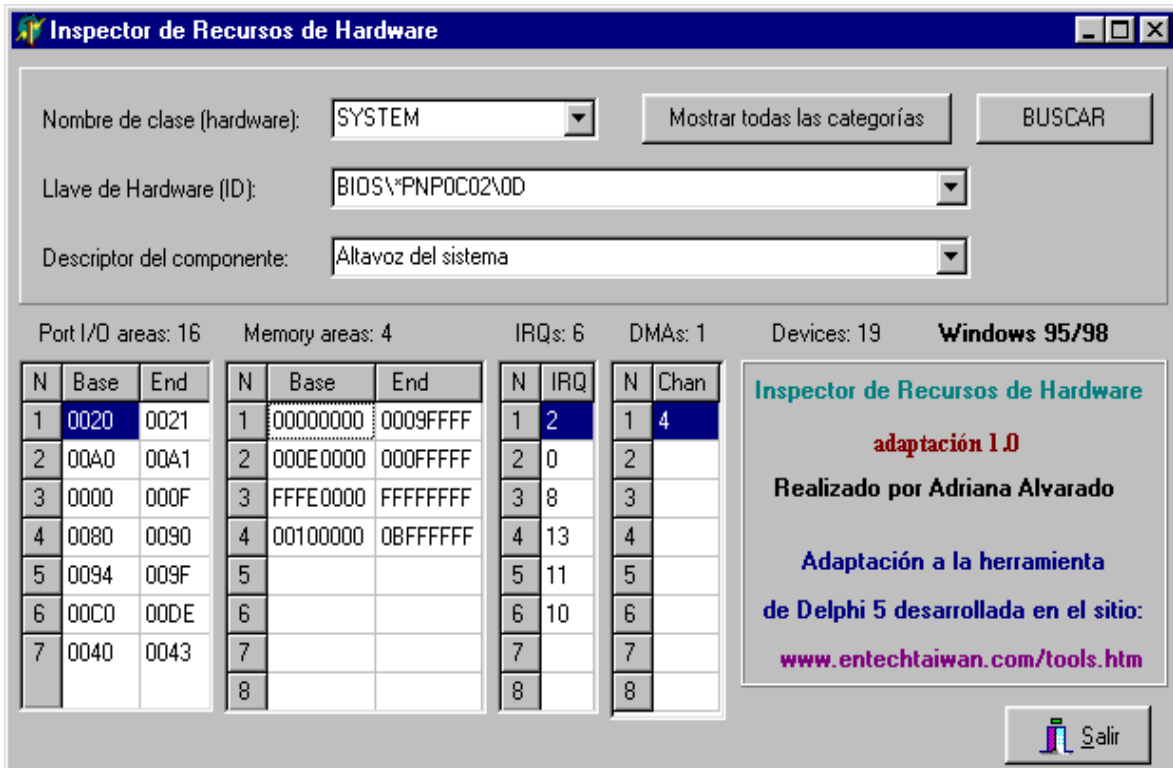


Figura A.4.5 Inspector de recursos de hardware del reloj nacional a partir del oscilador de cesio de 10 MHz.

### 1.2.4 Ayuda:

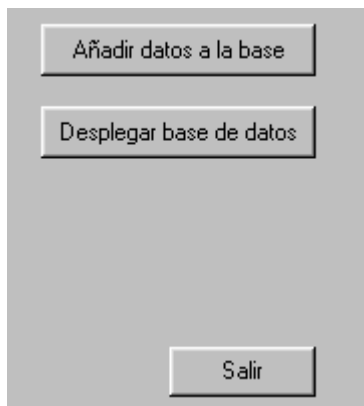
Al seleccionar esta opción, se abrirá una ventana de ayuda la cual contiene los principales aspectos del programa. En el capítulo “Ayuda y mal funcionamiento del sistema” se discutirá su funcionamiento.

## 1.3 Teclas de acceso rápido

Si el usuario así lo desea, puede acceder a la base de datos mediante una botón de acceso rápido dentro de la misma interfaz denominado “Desplegar base de

datos”. Si el usuario lo desea puede incluir algún tiempo fuera del intervalo definido por él para la toma de datos mediante la tecla “Añadir datos a la base”.

En el momento en el cual el usuario, seleccione esta opción, se generará un dialogo cuando desee salir de la aplicación. Esta opción abrirá una pequeña ventana que se muestra en la figura A.4.3, para que se ingrese la clave de salida. Una vez ingresada la contraseña, se hará una comparación entre el dato ingresado por el usuario con el especificado para funcionar como password. Si el dato es correcto, el programa finalizará su ejecución; en caso contrario mantendrá la ventana de clave de salida abierta. Si el usuario presiona la tecla Esc del teclado, la ventana de clave de salida se cerrará y el programa continuará ejecutándose. Las teclas de acceso rápido se observan en la figura 6



**Figura A.4.6** Teclas de acceso rápido del reloj nacional a partir del oscilador de cesio de 10 MHz.

## **Capítulo 2: Base de datos**

### **2.1 Tecla de acceso rápido “Cargar”**

Al seleccionar la opción “Desplegar base de datos” esta opción, se abrirá una ventana igual a la mostrada en la figura 4. Los datos se almacenan en un archivo basedatos.txt y se despliegan en la ventana dentro de la interfaz. Los datos se almacenan dentro del archivo de texto uno por línea, y el ciclo se repite indefinidamente hasta que la aplicación finalice.

Una vez presionada la tecla “Cargar”, la interfaz va a permitir visualizar los datos almacenados en el documento de texto, en forma de una lista, que es la que se visualiza en la figura A.4.7.

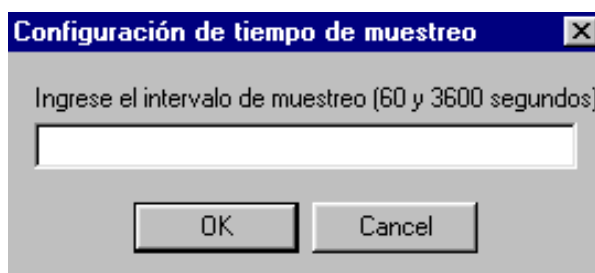
| Hora del Oscilador | Dato de la Hora del NIST | Porcentaje de error | Fecha    |
|--------------------|--------------------------|---------------------|----------|
| 11:01:36 AM        | 88 : 88: 88              | 0.0000%             | 09/01/03 |
| 11:02:07 AM        | 88 : 88: 88              | 0.0000%             | 09/01/03 |
| 11:03:19 AM        | 88 : 88: 88              | 0.0000%             | 09/01/03 |
| 11:03:47 AM        | 88 : 88: 88              | 0.0000%             | 09/01/03 |
| 11:04:16 AM        | 88 : 88: 88              | 0.0000%             | 09/01/03 |
| 11:24:04 AM        | 88 : 88: 88              | 0.0000%             | 09/01/03 |
| 11:40:35 AM        | 88 : 88: 88              | 0.0000%             | 09/01/03 |
| 10:31:34 AM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 10:32:02 AM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 10:32:33 AM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 10:33:01 AM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 10:33:30 AM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 12:55:38 PM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 12:56:41 PM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 12:57:09 PM        | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 6:25:13 PM         | 88 : 88: 88              | 0.0000%             | 18/01/03 |
| 7:13:05 AM         | 88 : 88: 88              | 0.0000%             | 20/01/03 |

Figura A.4.7 Base de datos del reloj patrón a partir del oscilador de cesio de 10MHz (Después de presionar la “Cargar”).

## 2.2 Opción de configuración de Tiempo de muestreo

El programa le ofrece al usuario la posibilidad de configurar el tiempo de toma y almacenamiento de los datos (intervalo entre 60 y 3600 segundos), el cual se puede acceder dentro de esta ventana desde su menú, al seleccionar la opción “Configurar el tiempo de toma de datos”. De esta forma aparecerá una ventana de diálogo similar a la mostrada en la figura A.4.8. Una vez que se ingrese el intervalo correcto el contador un contador para la toma de datos programado en la unidad interfaz.dcu tendrá el nuevo valor escogido, produciendo un nuevo conteo y un nuevo intervalo de datos. Si el intervalo que se ingresa no es el correcto el programa genera

una aviso y vuelve a pedir un tiempo de toma de datos dentro del intervalo establecido para la programación. Es importante destacar que este intervalo se determinó de acuerdo a las exigencias del ICE. Si el usuario presiona la tecla Esc, cancela la opción de configuración y el tiempo de toma de datos será el tiempo por defecto, o 60 segundos.



**Figura A.4.8 Opción configuración de tiempo de muestreo del menú de la base de datos.**

## **2.3 Ayuda**

Al seleccionar esta opción, se abrirá una ventana de ayuda la cual contiene los principales aspectos del programa.

## **Capítulo 3: Búsqueda de puertos: Estado de los puertos de la PC**

### **3.1 Aspectos generales del inspector de recursos**

El inspector de recursos que se muestra en la figura 5, es el encargado de ayudar al usuario en el caso de que el sistema no este funcionando correctamente, debido a que le permite al mismo realizar un diagnóstico de todos los dispositivos incluidos en la computadora donde se está ejecutando el programa. Esta ventana es una adaptación a una herramienta desarrollada en el sitio [www.entechtaiwan.com/tools.htm](http://www.entechtaiwan.com/tools.htm), misma que se adaptó para el sistema Windows 95 o 98 y que se muestra en la figura 5.

Cuando se presiona la opción “Mostrar todas las categorías”, se va a realizar una búsqueda a través del sistema operativo de Windows, de los recursos de hardware



disponibles dentro de la PC. En consecuencia se van a desplegar una serie de clases de hardware dentro de la casilla llamada “Nombre de clase (Hardware)”, que serán todos los recursos disponibles. Si se busca dentro de esta casilla la opción “SYSTEM”, y luego se presiona la tecla “Buscar”, deberá aparecer en la casilla “Descriptor de Componente”, el recurso del bus ISA hallado en la PC (Ver figura A.4.9). En caso contrario, el funcionamiento del sistema no se está dando a nivel de hardware. En consecuencia, este apartado servirá únicamente de referencia al usuario sobre el comportamiento del reloj.

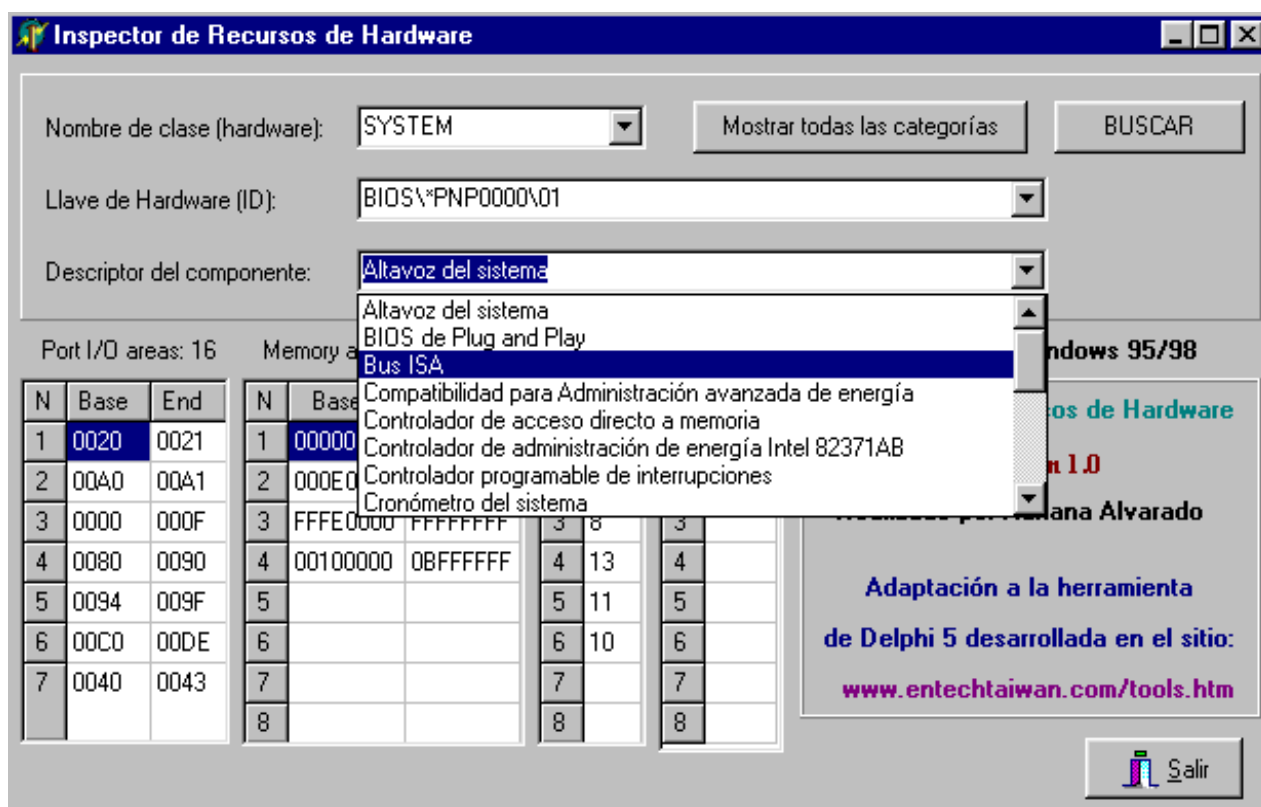


Figura A.4.9 Modo de búsqueda del bus ISA dentro del inspector de recursos de hardware.

## **Capítulo 4: Ayuda y mal funcionamiento del sistema**

### **4.1 Tecla de ayuda**

Al seleccionar esta opción, se abrirá una ventana de ayuda la cual contiene los principales aspectos del programa (instalación, manejo de datos, despliegue de datos en pantalla, base de datos, tiempo de muestreo, etc)

### **4.2 Mal funcionamiento del sistema**

Si el sistema del reloj tiene problemas en su funcionamiento realice los siguientes pasos para detectar las zonas de posible falla:

1. Seleccione en el menú principal, la opción “Búsqueda de puertos: Estado de puertos de la PC”; y siga el procedimiento de la sección 3.1. En caso de que no aparezca en la casilla “Descriptor de Componente”, el recurso del bus ISA, el funcionamiento del sistema no se está dando a nivel de hardware.
2. Si el Problema es a nivel de hardware, desmonte del zócalo el microcontrolador PIC16F877; y con un tester mida los posibles cortocircuitos entre el voltaje fuente y tierra o en alguna de las entradas utilizadas para su diseño. Para realizar lo anterior, utilice el circuito esquemático del sistema, y las hojas de datos del fabricante.
3. Si el PIC 16F877 posee cortos dentro del componente, reemplácelo por uno nuevo programando el código fuente en el mismo. Debe usar el lenguaje MPLAB, y un programador compatible.
4. Si no posee el programa MPLAB, puede descargarlo en forma gratuita de la página <http://www.microchip.com>

## **ANEXOS**

## ANEXO 1 Aspectos más relevantes del PIC 16F877



# PIC16F87X

## 28/40-pin 8-Bit CMOS FLASH Microcontrollers

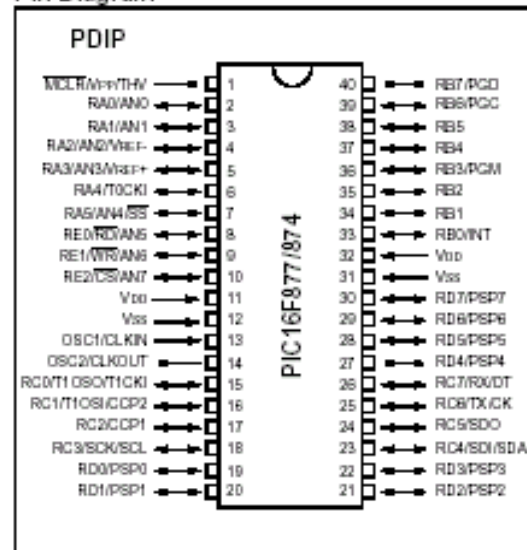
### Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

### Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM data memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and  
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC  
oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM  
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two  
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram



### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,  
can be incremented during sleep via external  
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period  
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master  
Mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver  
Transmitter (USART/SCI) with 9-bit address  
detection
- Parallel Slave Port (PSP) 8-bits wide, with  
external  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{CS}$  controls (40/44-pin only)
- Brown-out detection circuitry for  
Brown-out Reset (BOR)

## ANEXO 2 Aspectos más relevantes del LM7805



May 2000

### LM78XX Series Voltage Regulators

#### General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the out-

put, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

#### Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

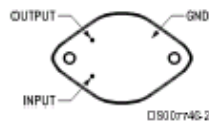
#### Voltage Range

|         |     |
|---------|-----|
| LM7805C | 5V  |
| LM7812C | 12V |
| LM7815C | 15V |

LM78XX Series Voltage Regulators

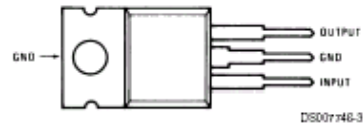
#### Connection Diagrams

Metal Can Package  
TO-3 (K)  
Aluminum



Bottom View  
Order Number LM7805CK,  
LM7812CK or LM7815CK  
See NS Package Number KC02A

Plastic Package  
TO-220 (T)



Top View  
Order Number LM7805CT,  
LM7812CT or LM7815CT  
See NS Package Number T03B