
CAPÍTULO IV . INTERFAZ

4.1. Introducción.

Actualmente en la mayoría de los sistemas electrónicos es común el uso de una computadora dentro del proceso de control. Esto es así, porque el software, más que el hardware, permite hacer cualesquier ajustes a los parámetros del sistema con el fin de obtener una respuesta deseada. Además, las computadoras también pueden efectuar funciones de supervisión, como la programación de numerosas aplicaciones requeridas.

Por otro lado, el uso de computadoras dentro del sistema tiene grandes ventajas entre ellas están:

- Reducción en el costo.
- Flexibilidad en respuesta a los cambios en el diseño.
- Incremento en la funcionalidad.
- Capacidad de capturar una gran cantidad de datos y procesarlos con rapidez.
- Capacidad de comunicación del software al hardware.

Todas estas ventajas y otras más hacen de una computadora una excelente opción para integrarla dentro de un sistema.

Tomando en cuenta lo anterior, dentro del proyecto propuesto se incluyó la utilización de una computadora que permitiera la comunicación con el sistema. Dando pie a la manipulación de los parámetros principales de entrada al sistema para obtener una respuesta deseada. Es decir, que mediante una interfaz entre la computadora y el FPGA, el usuario es capaz de modificar los estímulos que llegan como entradas al sistema para obtener una salida.

Esta comunicación entre la computadora y el FPGA se hizo mediante una interfaz por el puerto paralelo. La aplicación de este proceso se realizó valiéndose del lenguaje de programación Visual Basic. El proceso de comunicación mediante la interfaz, así como su diseño en programación de Visual Basic y VHDL para este sistema se explicarán en este capítulo.

4.2. Introducción al lenguaje de programación Visual Basic.

A medida que pasan los años los lenguajes de programación han ido evolucionando dando lugar a nuevos métodos de programación que facilitan al programador el diseño de aplicaciones complejas. Este es el caso del método de programación orientado a objetos (eventos), que permite realizar aplicaciones en un entorno gráfico, facilitando así al usuario su utilización y al programador su diseño, tal es el caso de los programas típicos de Windows como Word, Excel, PowerPoint, etc.

Estos programas se llaman *orientados a eventos* porque esperan a que el usuario realice algún evento, ya sea con el mouse o son el teclado para realizar alguna función. Tomando como evento una acción que realiza el usuario hacia un objeto.

Visual Basic es un lenguaje de programación orientado a eventos, lo cual quiere decir que es una excelente herramienta de programación. La razón principal de esto, es que se ha diseñado para facilitar el desarrollo de aplicaciones en un entorno gráfico (GUI, Graphical User Interface) como Windows 95/ 98 / NT [11]. Con este lenguaje, es posible escribir aplicaciones con elementos gráficos, como ventanas, cuadros de diálogo, menús, barras de herramientas, botones de opción, cuadros de lista, informes y páginas Web, entre otros.

En Visual Basic se le llama objeto a todo lo que se ve en una ventana típica de Windows; los objetos son por ejemplo un botón de comando, una caja de texto, una

imagen, en general todo objeto visible que se pueda apreciar en la pantalla. Y se les llama objetos porque cada uno de ellos posee propiedades, eventos y métodos [12].

El entorno general del programa Visual Basic se conforma de los siguientes elementos básicos:

Uno de los principales es la ventana editor del proyecto mostrada en la figura 4.1.

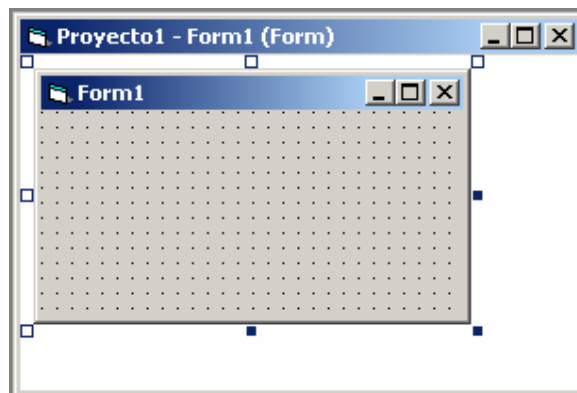


Figura 4.1. Ventana Editor de Proyecto

Esta ventana permite editar los componentes que se utilizan para la aplicación, visualizando las ventanas de código, los módulos y los informes de datos. Y dentro de esta, se encuentra la ventana de Formulario que es el área donde se diseña la interfaz gráfica, es decir que permite al diseñador insertar los objetos o elementos gráficos como botones, imágenes, cuadros de texto, cuadros de lista, etc., necesarios para crear el ambiente gráfico de la aplicación.

Por otro lado, tenemos el cuadro de herramientas encontrado en la figura 4.2.



Figura 4.2. Cuadro de Herramientas Visual Basic 6.0

Este cuadro presenta todos los controles para diseñar una aplicación, como cuadros de texto, etiquetas, cuadros de lista, botones de comando, etc. Estos controles se pueden añadir a la ventana de formulario para crear la interfaz gráfica que se desee.

Y por último, otro elemento básico que se tiene es la Ventana de Propiedades mostrada en la figura 4.3.

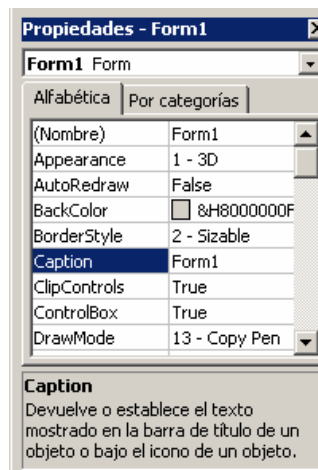


Figura 4.3. Ventana de Propiedades de Visual Basic.

La ventana de propiedades de Visual Basic muestra la lista de propiedades que pueden asignarse a un objeto o control activo. En la parte inferior, se encuentra una breve descripción de la propiedad especificada o seleccionada.

Aunque Visual Basic contiene muchos otros elementos con diferentes aplicaciones, con el objeto de comprender en general, la realización del software para la interfaz por medio del puerto paralelo del sistema diseñado en esta tesis, solo se describirán estos elementos básicos.

4.3 Estructura gráfica de la Interfaz del Puerto Paralelo del Sistema.

El primer paso para lograr la comunicación por el puerto paralelo entre el sistema y la computadora fue crear el ambiente gráfico de la aplicación. Este se creó basándose en los elementos básicos de Visual Basic, mencionados en el apartado anterior. Tomando como primer punto la generación de un proyecto, en cuya ventana de formulario se fueron ingresando diferentes objetos o controladores. En este caso, se insertaron botones de comando (Cargar, Salir y Enviar) y cajas de texto con sus respectivos nombres (Voltaje1, Voltaje2, Frecuencia y Desfasamiento), obtenidas del cuadro de herramientas. Y en donde a cada control se le asignaron sus respectivas propiedades de acuerdo con su función.

Realizando lo anterior, se llegó al siguiente entorno gráfico mostrado en la figura 4.4.

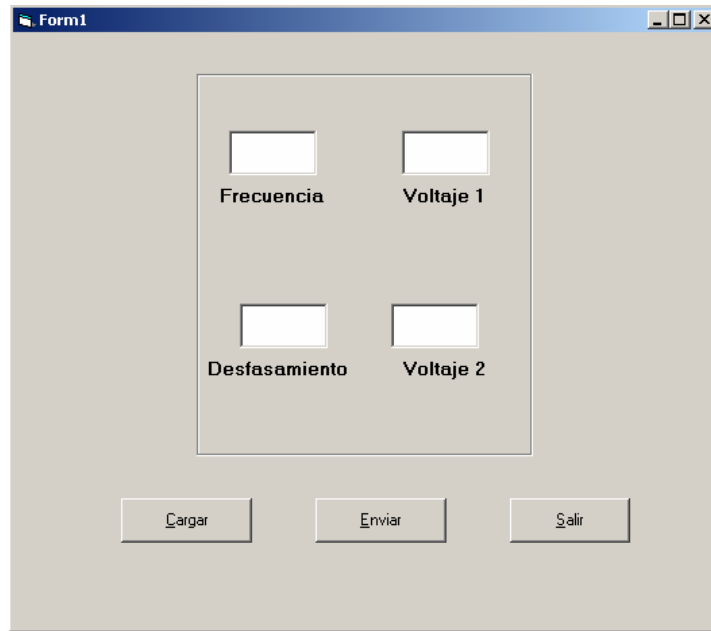


Figura 4.4. Entorno gráfico de la Interfaz del Puerto Paralelo

4.4. Programación de Interfaz del Puerto Paralelo en Visual Basic.

Visual Basic pese a su potencial y facilidad de programación no dispone en forma directa de instrucciones que permitan leer o escribir datos al puerto paralelo de la computadora.

Para hacer posible el acceso al puerto paralelo es indispensable la incorporación de una DLL que son librerías de enlace dinámico, uno de los elementos principales del sistema operativo Windows 9X. Estas librerías contienen archivos ejecutables independientes que están formados por funciones que pueden ser llamadas por otros programas e incluso por otras DLL.

Las DLL no pueden ser ejecutadas de forma independiente, más bien, entran en acción en el momento en que un programa llama a una de las funciones de la librería. Por lo que estas librerías solo entran en acción en el momento en que es requerido, en el tiempo de ejecución. En este caso, el programa que llama a las funciones de la librería dinámica es Visual Basic.

Sin embargo, una vez que se tiene la librería DLL instalada dentro del sistema operativo, lo que se requiere para realizar la operación de escritura en el puerto paralelo son dos parámetros: la dirección del puerto paralelo y el dato que se desea escribir al puerto. Lógicamente, el valor del dato deberá estar dentro del rango de 0 a 255 equivalentes a 8 bits que son los que puede mandar el puerto paralelo.

En general, la dirección del puerto paralelo LPT1 es en hexadecimal de 0x378h, que en números decimales equivale al valor de 888. Por lo tanto, ya teniendo ambos parámetros la instrucción utilizada para escribir al puerto paralelo en Visual Basic es:

```
Result = Escritura(888, Valor)
```

declarando las variables, en este caso *Result* como un número booleano y el *valor* como entero.

En la programación realizada para la interfaz del sistema con la computadora, en realidad, de los 8 bits que se mandaban cada cierto tiempo los cuatro más significativos correspondieron a 4 bits del dato, mientras que los otros 4 bits del valor se destinaron como bits de control para la recepción del dato en el FPGA. De modo, que fue indispensable hacer la partición del dato y mandar de 4 bits en 4 bits, hasta completar los 4 valores necesarios (V1, V2, N, A) correspondientes a las entradas del sistema. El código fuente se encuentra en el Apéndice C, en el disco compacto adjunto.

4.5. Programación de la Recepción de Datos en el FPGA

Para que exista una comunicación adecuada entre la computadora y cualquier otro dispositivo, ambos deben de estar programados, uno para transmitir los datos y el otro para esperar la recepción.

En este caso, el sistema diseñado requiere que la computadora sea el dispositivo que escriba los datos que el usuario ingrese, al puerto paralelo y los transmita hacia el

FPGA. A su vez, el dispositivo de recepción que es el FPGA, tendrá que estar programado para esperar los datos de la computadora y guardarlos en los registros adecuados para que el sistema los procese.

Anteriormente, ya se mencionó la parte que corresponde al diseño y programación de la etapa de transmisión de la computadora mediante Visual Basic. Ahora bien, la etapa de recepción que le corresponde al FPGA se programó mediante VHDL.

Esta etapa esta organizada por módulos como se muestra en la figura 4.5.

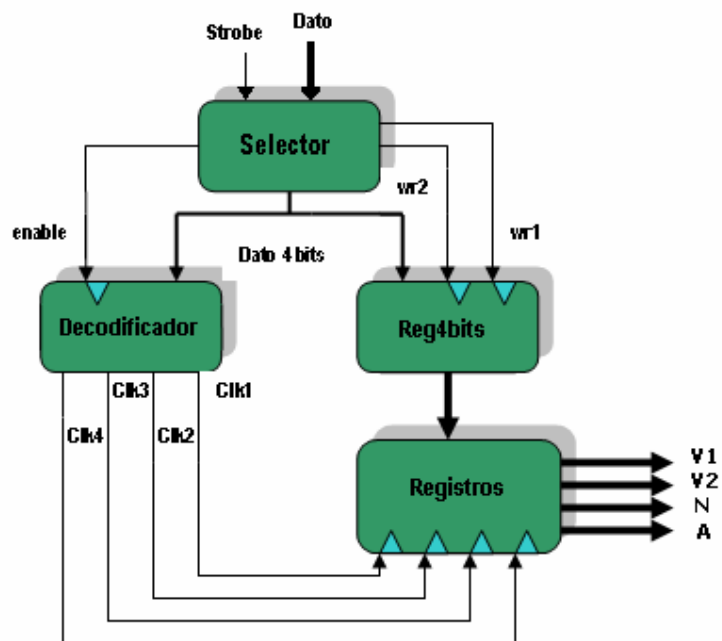


Figura 4.5. Diagrama a bloques de programa de recepción en VHDL

El módulo de *Selector* se encarga de dividir los 8 bits que vienen por el puerto paralelo, dejando pasar los 4 bits más significativos que corresponden al dato hacia el *Decodificador* y a *Reg4bits*. Mientras que utiliza los 4 bits menos significativos con una lógica combinacional para generar las señales de reloj *enable*, *wr1*, *wr2*.

La lógica combinacional interna al módulo de *Selector* se muestra en la figura 4.6.

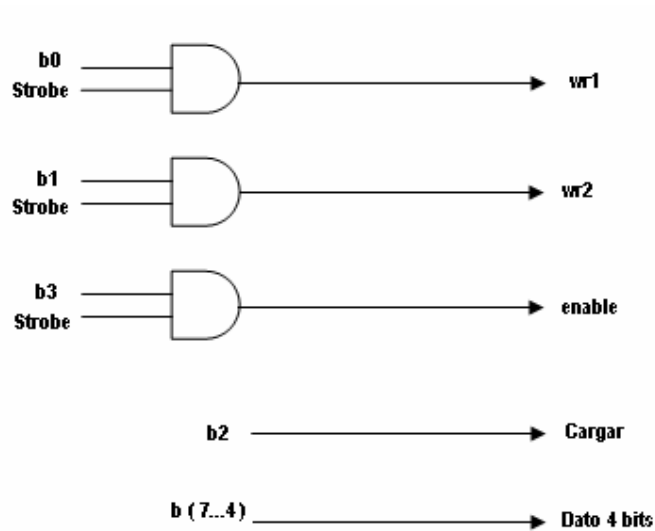


Figura 4.6. Lógica combinacional interna de *Selector*.

De manera que dependiendo de los bits de control mandados es como se cargará primero el dato_H (con wr1) y luego el dato_L (con wr2). Finalmente, ya que el dato completo se encuentra en el bus de datos que llega a los *Registros* (4 específicamente) entonces el decodificador recibirá un dato de 4 bits que indicará cual de los cuatro relojes activará para cargar el valor en el registro adecuado.

Esta rutina se repetirá hasta terminar de cargar los cuatro datos completos de entrada al sistema. Y por último, el bit 2 cargará los valores en los módulos adecuados dentro del sistema en el FPGA, cuando ya se recibieron todos los 4 datos de entrada.

El programa realizado en VHDL que permite el uso de la interfaz por el puerto paralelo se encuentra en el Apéndice C, en el disco compacto adjunto. Por otro lado, en el Apéndice D se añade un manual para el usuario, que facilita el uso de la fuente de voltaje programable desde la aplicación gráfica en la computadora.