

CAPÍTULO 4

SOFTWARE

4.1 Microsoft Visual Basic

Se recurrió a la Versión de Visual Basic 6.0 de Microsoft Corp. para poder hacer la interfaz gráfica de control del proyecto y por su accesibilidad.

Este lenguaje de programación de alto nivel fue utilizado por la facilidad de desarrollar aplicaciones complejas en muy poco tiempo, buena eficiencia para tareas sencillas e incorporar elementos del entorno de Windows que manejan el uso de puertos seriales, la comunicación recepción - transmisión asíncrona y temporizadores. Aunque en comparación con otros lenguajes como C o C++, es de menor velocidad. Para la aplicación que se le da, en este caso crear un entorno de recepción y transferencia de datos para el control de la planta es bastante accesible y eficaz.

Este lenguaje es llamado también de cuarta generación, esto quiere decir que se pueden recrear un gran número de tareas sin escribir códigos, simplemente generar eventos. Además está basado y orientado a objetos, utilizando propiedades y métodos para su manipulación [7].

4.1.1 Interfaz en Visual Basic

Se incorporó esta herramienta para la comunicación, en específico la rutina de inicio de la planta o sistema mecánico, la transferencia de información de las mediciones tomadas por el sensor y la generación de una base de datos. En base a una interfaz gráfica donde se pueden realizar diversas acciones principalmente la observación de las mediciones y la inicialización del sistema.

Se pudo haber desarrollado en otros lenguajes de programación como C o C++ y Java, pero por los objetivos y su rápido aprendizaje fue conveniente utilizar Visual Basic 6. Ya que cuenta con una plataforma de programación con elementos de ejecución en tiempo real y a pasos, de prueba y realización de archivos ejecutables sin el uso de programas externos.

4.1.2 Diseño de control

Se creó una plataforma visual que está compuesta de una ventana principal que dispone de la selección del puerto a usar, crear la conexión, cerrar la conexión, cuadro de texto para transmisión de información y botón de enviar, cuadro de texto de la información recibida, botón de limpiar la información recibida y barra de menú salir del programa.

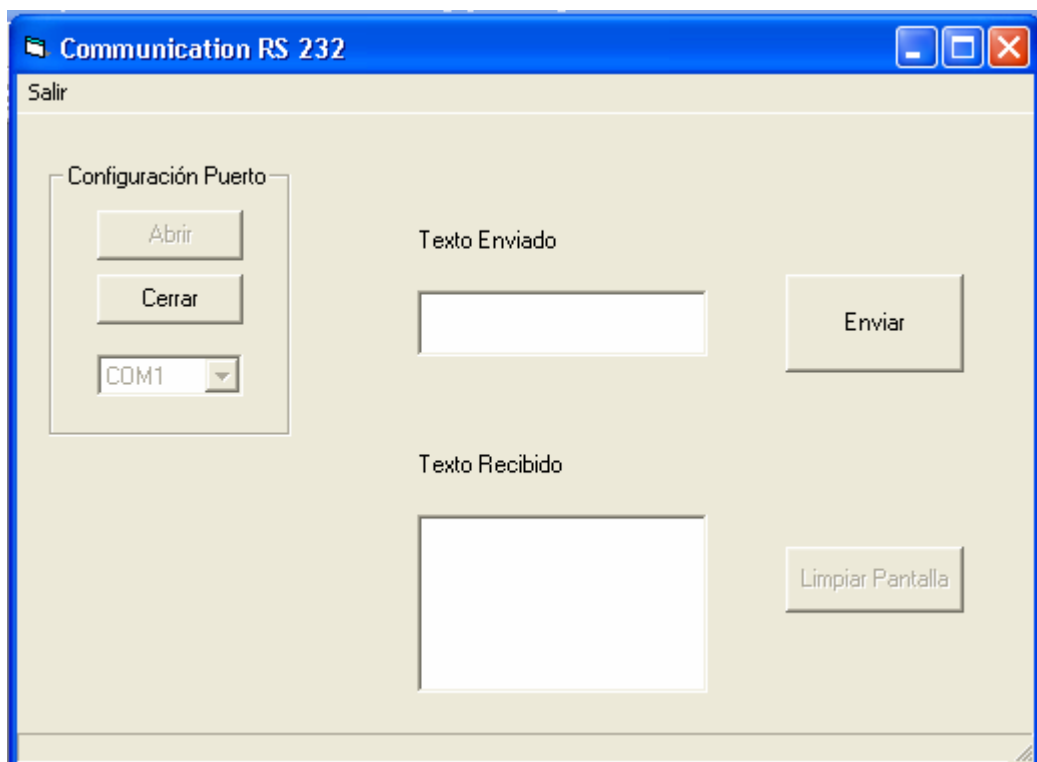


Figura 4.15 Diagrama de la Interfaz Gráfica de Visual Basic

Los eventos que se crearon se resumen en:

- Evento de carga del programa realiza la cancelación de todos los menús de inicio y de los puertos, COM1 y COM2, hasta seleccionar el puerto correspondiente.
- Seleccionar puerto que consiste en indicar qué puerto se usará y abrirlo para después permitir al usuario considerar las demás opciones o no se podrá hacer absolutamente nada.
- Evento de abrir el puerto que consiste en habilitar la función de botón y texto de enviar y recibir, aunado al botón limpiar.
- Evento de cerrar la aplicación con el que se cierran todas las opciones de la ventana principal.
- Evento de enviar que hace una comparación de si el puerto se encuentra abierto mande todo lo que se encuentra en el cuadro de texto de enviar.
- Evento limpiar que consiste en dar la apariencia de texto en blanco en la zona de texto recibido.
- Evento de Comunicación encendida que realiza la mayor parte del archivado y desplegado de los datos. Cuenta con una rutina que se mantiene, si la propiedad del buffer de entrada es igual a un dato o caracter de cualquier tipo. Desplegando en el cuadro de texto de recibido un dato nuevo cada vez que se recibe y calculando mediante ecuaciones características para cada rango de medición, la medición de la altura desplegada en centímetros.
- Después es almacenada en un archivo de texto junto con las coordenadas faltantes X e Y , de manera matricial y llevando una sistematización particular. Es decir, la variable Y de 0 a 99 y de regreso, y para la variable X de 0 a 99, en coordinación con las mediciones adquiridas. Esta rutina es llamada de nuevo,

tantas veces sea necesaria por la introducción de las mediciones. Se generó el archivo de esta manera:

Z	Y	X
Z ₀ ...	0...	0...
Z ₉₉	99	0
Z ₉₉ ...	99...	1...
Z ₀	0	1
....99

Tabla 4.2 Matriz Creada en Archivo

Para que no interfirieran algunas mediciones mal tomadas por el sensor se creó un algoritmo de control de error, que si disminuye la medición del parámetro inicial de cero, éste se quedará en cero. En pocas palabras si el sensor sale del rango utilizado por la ecuación, establecido por la gráfica característica, se normaliza a cero siempre y cuando sea para el dato mínimo cero, de esta manera:

```
If Buffer <= 109 Or Buffer >= 222 Then Z = 0 End If
```

La ecuación que caracteriza de las mediciones del sensor fue adquirida por el proveedor SHARP y la compañía Acroname Robotics. Fue acondicionada y caracterizada de forma que sea compatible con el sistema.

Esta fue la ecuación característica del sensor:

$$\text{Altura en Centímetros} = (19 / (\text{Tan}(\text{Buffer} * 0.025 * 3.1415926554 / 180))) - 100$$

Y fue modificada a conveniencia de esta forma:

$$Z = (((19 / (\text{Tan}(\text{Buffer} * 0.025 * 3.1415926554 / 180))) - 100) / 10$$

Se dividió entre diez porque el resultado que daba al realizar las operaciones con la ecuación original no proporcionaba los valores que se requieren para esta aplicación.

Después se le incorporó una compensación para que fuera el número mayor de la tabla igual a cero y el número menor igual a la altura máxima. La ecuación nos realizaba una función definida dentro del rango de 10 a 40 centímetros y la función que se requería debía ser de 0 a 30 centímetros. Pero invertida, 40 debía ser 0 y 10 debía ser 30, como se indica en la siguiente tabla.

Resultado de la Función Original dividida entre 10	Resultado de la Función Modificada
40 cm	0 cm
30 cm	10 cm
20 cm	20 cm
10 cm	30 cm

Tabla 4.3 Resultados de la Función Original y Modificada

Función Modificada:

$$Z = \text{Abs}(\left(\left(\left(19 / (\text{Tan}(\text{Buffer} * 0.025 * 3.1415926554 / 180))\right) - 100\right) / 10\right) - 29.9)$$

Ahora bien, como se tiene un rango muy grande y una función característica que no se apega a las mediciones en especial en algunos puntos. Se decidió por delimitar el rango de operación y dividir en diferentes ecuaciones para ciertos rangos. Todos los valores fueron acondicionados por medio de extenuantes mediciones.

El primer rango se especificó de 0 a 5.9 cm. que corresponde a valores del sensor de 110 a 130, aplicado de esta manera:

If Buffer >= 110 And Buffer <= 130 Then

$$Z = \text{Abs}(\left(\left(\left(19 / (\text{Tan}(\text{Buffer} * 0.025 * 3.1415926554 / 180))\right) - 100\right) / 10\right) - 29.9)$$

End If

El segundo rango se definió de 6 a 13.9 cm. que corresponde a los valores del sensor de 131 a 172.

If Buffer >= 131 And Buffer <= 172 Then

$$Z = \text{Abs}(\left(\left(\left(19 / (\text{Tan}(\text{Buffer} * 0.025 * 3.1415926554 / 180))\right) - 100\right) / 10\right) - 29.3) - 0.7$$

End If

Tercer rango de 14 a 15.9 cm.,

If Buffer >= 173 And Buffer <= 190 Then

$$Z = \text{Abs}(\left(\left(\left(19 / (\text{Tan}(\text{Buffer} * 0.025 * 3.1415926554 / 180))\right) - 100\right) / 10\right) - 29.3) - 1$$

End If

Y por último de 16 a 30.

If Buffer >= 191 And Buffer <= 221 Then

$$Z = \text{Abs}(\left(\left(\left(19 / (\text{Tan}(\text{Buffer} * 0.025 * 3.1415926554 / 180))\right) - 100\right) / 10\right) - 29.3) - 2.2$$

End If

El máximo valor que puede ser capturado con una buena precisión y margen de error en las ecuaciones, es 15 cm., esto comprobado mediante pruebas. Ya que valores más allá de estos no cuentan con rango de operación.

4.2 MATLAB

Las pruebas finales de graficación estuvieron basadas en la herramienta de programación, graficación y manipulación matemática de MATHWORKS INC llamada MATLAB versión 7.0.

MATLAB cuenta con un sinfín de funciones para el análisis de señales, cálculos numéricos, graficación, librerías y paquetes integrados, así como también la interacción con otros programas como LABVIEW.

Esta paquetería sirvió para la graficación de la totalidad de los puntos medidos y anteriormente almacenados en forma matricial, debido a que cuenta con diversas funciones y análisis. El archivo antes mencionado es cargado y graficado para recrear el objeto.

Como método de disminución de errores en las mediciones realizadas a través del sensor por los factores externos, se utilizaron primero funciones que relacionan la totalidad de las mediciones en forma de superficie y no en vectores simples. Después, la utilización de la función probabilística de interpolación para los puntos generados a la superficie y por último la graficación de forma volumétrica de estos.

4.2.1 Principios de Operación de las Funciones

Mesh

Función que crea superficies paramétricas especificadas por los vectores X , Y y Z . Esta función grafica en forma de malla la superficie proporcional a los puntos, generando los diferentes volúmenes o superficies por colores [14].

Meshgrid

Función que transforma el dominio especificado por los vectores X y Y , utilizados para evaluar funciones de dos variables y diagramas tridimensionales de Mesh o Surface. Las filas de salida X son copias del vector x y las columnas de la salida Y son copias del vector y [14].

Surf

Función que crea superficies paramétricas especificadas por X , Y , Z . Crea una superficie sombreada tridimensional de los componentes de z en la matriz Z , usando $x = 1:n$ y $y = 1:m$, donde $[m,n] =$ dimensión de Z . La altura es evaluada sobre una rejilla geoméricamente rectangular. Los colores especifican los datos de la altura superficial [14] y donde m y n representan las dimensiones máximas de la matriz

Griddata

Función que genera los vectores uniformemente espaciados interpolando la superficie en puntos específicos cerca de (X_i, y_i) para producir Z_i . La superficie pasa siempre a través de los puntos de referencia [14]. Donde I e i determinan un número finito de puntos y X , y y Z representan los ejes coordenadas diferentes, las mayúsculas disímiles de las minúsculas.