Implementación de un VI

Esta lección le enseña cómo implementar código en LabVIEW. Se explicará cómo diseñar una interfaz de usuario, elegir un tipo de datos, documentar su código, usar estructuras de bucles como los bucles While y For, añadir temporización por software a su código, mostrar sus datos como una curva y tomar decisiones en su código utilizando una estructura Case.

Temas

A. Diseño del panel frontal
B. Tipos de datos de LabVIEW
C. Documentación de código
D. Bucles While
E. Bucles For
F. Temporización de VIs
G. Transferencia de datos iterativa
H. Representación de datos
I. Estructuras Case

A. Diseño del panel frontal

En la fase de diseño del método de desarrollo de software se identifican las entradas y salidas del problema. Esta identificación conduce directamente al diseño de la ventana del panel frontal.

Puede recuperar las entradas del problema con los siguientes métodos:

- adquiriendo desde un dispositivo de adquisición de datos o un multímetro
- leyendo directamente desde un fichero
- manipulando controles

Puede ver las salidas del problema con indicadores, como gráficos o LEDs o registrar las salidas en un fichero. También puede extraer datos de un dispositivo con la generación de señales. Las lecciones de adquisición de datos, generación de señales y registro en ficheros se tratarán después en este curso.

Diseño de controles e indicadores

Al elegir controles e indicadores, deben ser apropiados para la tarea que desee realizar. Por ejemplo, cuando quiera determinar la frecuencia de una onda sinusoidal, elija un control de disco; cuando desee ver la temperatura, elija un indicador de termómetro.

Etiquetas y títulos

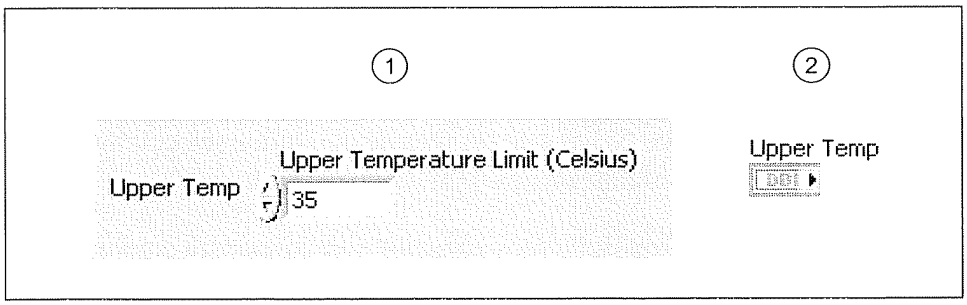
Etiquete con claridad los controles e indicadores. Estas etiquetas ayudan a los usuarios a identificar el objetivo de cada control e indicador. Asimismo, el etiquetado claro ayuda a documentar su código en el diagrama de bloques. Las etiquetas de controles e indicadores se corresponden con los nombres de los terminales del diagrama de bloques.

Los títulos ayudan a describir el control de un panel frontal. Los títulos no aparecen en el diagrama de bloques. El uso de títulos permite documentar la interfaz del usuario sin abarrotar el diagrama de bloques con texto adicional. Por ejemplo, en la estación meteorológica debe incluir un límite superior para el nivel de temperatura. Si la temperatura supera este nivel, la estación meteorológica indicará una advertencia de golpe de calor. Podría llamar a este control `Upper Temperature Limit (Celsius)`. Sin embargo, esta etiqueta ocuparía espacio innecesario en el diagrama de bloques. Use un título para el control `Upper Temperature Limit (Celsius)` y la etiqueta para crear una descripción más breve para el diagrama de bloques, como `Upper Temp`.

Opciones de controles e indicadores

Puede configurar valores predeterminados para los controles. La figura 4-1 muestra un valor predeterminado de 35 °C. Configurando un valor predeterminado puede adoptar un valor razonable para un VI si el usuario no configura otro valor durante el tiempo de ejecución. Para configurar el valor predeterminado, siga estos pasos:

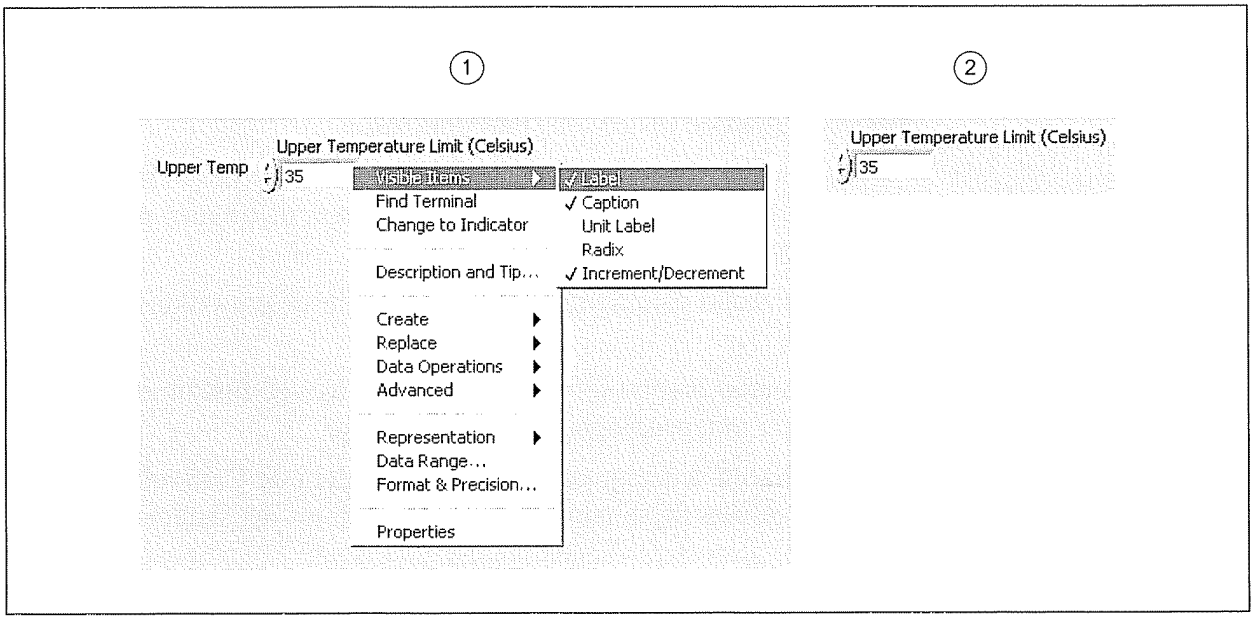
1. Escriba el valor deseado.
2. Haga clic con el botón derecho en el control y seleccione **Data Operations»Make Current Value Default** en el menú contextual.



1 Ventana del panel frontal 2 Diagrama de bloques

Figura 4-1. Configuración de valores predeterminados

También puede ocultar y ver elementos en controles e indicadores. Por ejemplo, en la figura 4-2 puede ver el título y la etiqueta. No obstante, sólo necesita ver el título. Para ocultar la etiqueta, haga clic con el botón derecho en el control y seleccione **Visible Items»Label**, como se ve en la figura 4-2.



1 Antes de ocultar la etiqueta 2 Después de ocultar la etiqueta

Figura 4-2. Ocultar una etiqueta del panel frontal

Uso del color

El uso adecuado del color puede mejorar el aspecto y la funcionalidad de su interfaz de usuario. Sin embargo, si usa demasiados colores pueden chocar colores que den un aspecto demasiado ocupado y distraído a la interfaz de usuario.

LabVIEW incluye un selector de color que puede ayudar a elegir los colores adecuados. Seleccione la herramienta de color y haga clic con el botón derecho en un objeto o espacio de trabajo para ver el selector de color. La parte superior del selector de color contiene un espectro en escala de grises y un cuadro que puede usar para crear objetos transparentes. El segundo espectro contiene colores apagados, ideales para fondos y objetos del panel frontal. El tercer espectro contiene colores ideales para resaltar. Si mueve el cursor verticalmente desde los colores de fondo a los colores de resaltar, le ayudará a seleccionar los colores apropiados para resaltar un color de fondo específico.

Los siguientes consejos son útiles para combinar colores:

- Use los colores predeterminados de LabVIEW. LabVIEW sustituye colores de forma similar a como sustituye fuentes. Si uno de los colores del VI no se encuentra, LabVIEW lo sustituye por el más similar. También puede utilizar los colores del sistema para adaptar el aspecto de una ventana del panel frontal a los colores del sistema de cualquier ordenador que ejecute el VI.
- Empiece con los grises. Seleccione uno o dos tonos de gris y elija colores para resaltar que contrasten bien con el fondo.
- Añada colores de resaltado con moderación (en curvas, botones de abortar y quizá las miniaturas del deslizador) para configuraciones importantes. Los objetos pequeños necesitan colores más fuertes y más contraste que los objetos más grandes.
- Use diferencias de contraste más que diferencias de color. A los usuarios daltónicos les cuesta discernir objetos cuando se distinguen por el color más que por el contraste.
- Use el espaciado para agrupar objetos en lugar de por colores que combinen.
- Puede aprender mucho sobre el color en paneles de instrumentos de hardware independiente, mapas y revistas.
- Elija objetos en la categoría **System** de la **Paleta de Controles** si desea que los controles de su panel frontal usen colores del sistema.

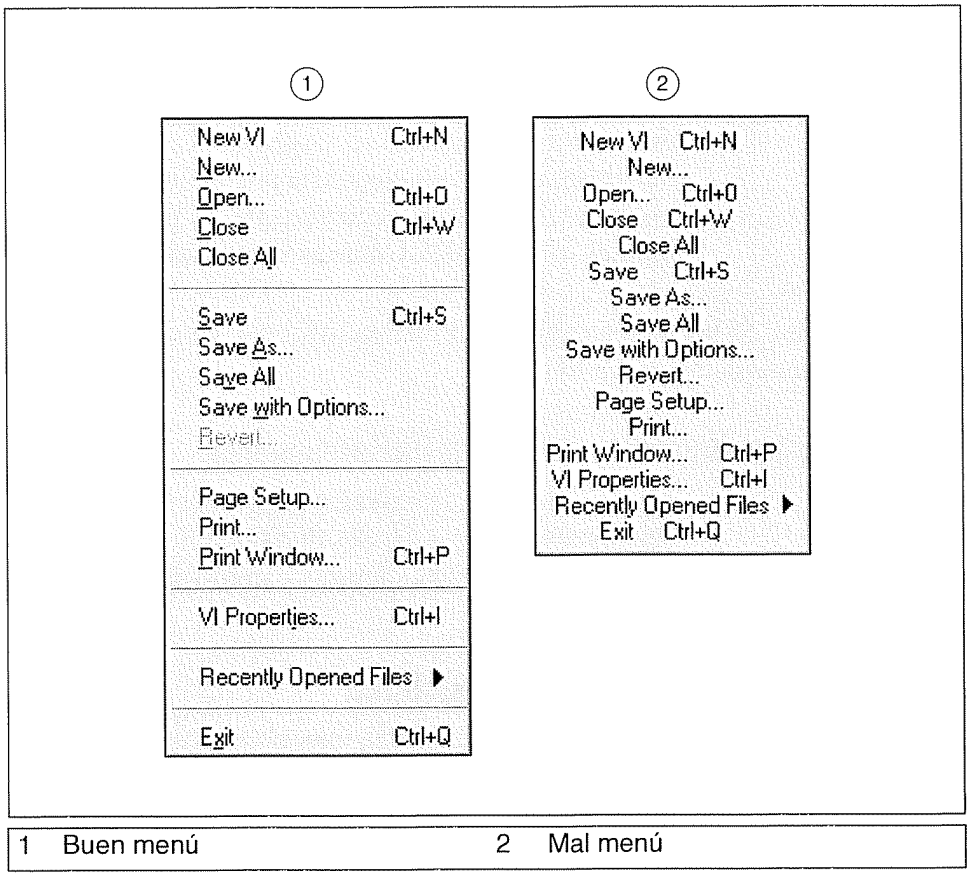
Espaciado y alineación

El espacio en blanco y la alineación probablemente sean las técnicas más importantes para agrupar y separar. Cuantos más elementos pueda encontrar el ojo en una línea, más limpia y más cohesiva parecerá la organización. Cuando los elementos están en una línea, el ojo sigue ésta de izquierda a derecha o de arriba a abajo. Esto se relaciona con la dirección de la escritura. Aunque algunas culturas ven los elementos de derecha a izquierda, casi todas lo hacen de arriba a abajo.

Cuando diseñe el panel frontal, piense en cómo los usuarios interaccionan lógicamente con el VI y agrupan controles e indicadores. Si varios controles están relacionados, añada un borde decorativo a su alrededor o colóquelos en un cluster.

No coloque objetos demasiado juntos. Deje algún espacio en blanco entre los objetos para facilitar la lectura del panel frontal. El espacio en blanco también evita que los usuarios hagan clic en el control o botón equivocado.

Observe cómo las líneas divisorias entre las secciones de menú de la figura 4-3 ayudan a buscar los elementos rápidamente y a fortalecer la relación entre los elementos de las secciones.



1 Buen menú 2 Mal menú

Figura 4-3. Ejemplos de buen y mal menú

No coloque objetos encima de otros. Incluso el cubrir parcialmente un control o indicador con una etiqueta u otro objeto ralentizará las actualizaciones de la pantalla y puede hacer parpadear el control o indicador.

Texto y fuentes

El texto es más fácil de leer y la información se entiende mejor cuando se encuentra ordenado correctamente. Use las fuentes predeterminadas de LabVIEW. Por ejemplo, LabVIEW define sus fuentes internas como las fuentes predeterminadas del sistema. Cuando mueve VIs entre plataformas, LabVIEW actualiza automáticamente sus fuentes internas para que coincidan con la fuente predeterminada del sistema de la plataforma actual. Además, si intenta abrir un VI que usa una fuente no disponible, LabVIEW la sustituirá por la más parecida. Por ejemplo, si intenta ver un VI que usa la fuente Arial en un ordenador que no la tiene, LabVIEW la sustituirá por una similar de las instaladas.

Si usa demasiados estilos de fuente, su ventana del panel frontal puede parecer saturada y desorganizada. Use en su lugar dos o tres tamaños distintos de la misma fuente. Las fuentes tipo Serif ayudan a reconocer palabras enteras desde lejos. Si utiliza más de un tamaño de una misma fuente, los tamaños deben diferenciarse bien. De lo contrario, parecerá un error. Igualmente, si usa dos fuentes distintas, diferéncielas bien.

Diseñe su interfaz de usuario con fuentes más grandes y más contraste para estaciones industriales. El resplandor del alumbrado o la necesidad de leer información desde la distancia pueden dificultar la lectura de las fuentes normales. Asimismo, recuerde que las pantallas táctiles suelen requerir fuentes más grandes y más espaciado entre los elementos de selección.

Nota Si las fuentes no existen en una máquina, las que se sustituyan pueden hacer que la interfaz aparezca torcida.

Consejos y herramientas de la interfaz de usuario

Algunas de las herramientas integradas de LabVIEW para crear ventanas de paneles frontales fáciles de usar son: controles del sistema, controles de fichas, decoraciones, menús y cambio de tamaño automático de objetos del panel frontal.

Controles del sistema

Una técnica común de la interfaz de usuario es mostrar cuadros de diálogo en el momento apropiado para interactuar con el usuario. Puede hacer que un VI actúe como un cuadro de diálogo seleccionando **File»VI Properties**, seleccionando la categoría **Window Appearance** y después la opción **Dialog**.

Use los controles e indicadores del sistema situados en la paleta **System** en los cuadros de diálogo que cree.Los controles del sistema cambian de aspecto en función de qué plataforma ejecute el VI. Si ejecuta el VI en una plataforma distinta, los controles del sistema adaptan su color y aspecto para coincidir con los controles del cuadro de diálogo estándar de esa plataforma.

Los controles del sistema suelen descartar todos los colores salvo el transparente. Si está integrando un gráfico o un control que no es del sistema en las ventanas del panel frontal, combínelos ocultando algunos bordes o seleccionando colores parecidos a los del sistema.

Controles de fichas

Los instrumentos físicos suelen tener buenas interfaces de usuario. Imite en gran medida sus principios de diseño, pero use controles más pequeños o eficientes, como controles de anillo o de ficha, cuando sea apropiado. Use controles de ficha para superponer controles e indicadores del panel frontal en una zona más pequeña.

Para añadir páginas adicionales a un control de ficha, haga clic con el botón derecho en una ficha y seleccione **Add Page Before** o **Add Page After** desde el menú contextual. Vuelva a etiquetar las fichas con la herramienta de etiquetado y coloque objetos del panel frontal en las páginas correspondientes. Los terminales de estos objetos están disponibles en el diagrama de bloques, al igual que los terminales de otros objetos del panel frontal (excepto las Decoraciones).

Puede cablear el terminal de control enumerado del control de la ficha con el selector de una estructura Case para producir diagramas de bloques más limpios. Con este método se asocia cada página del control de ficha con un subdiagrama de la estructura Case. Puede colocar los terminales de controles e indicadores de cada página del control de ficha (y los nodos y cables del diagrama de bloques asociados con esos terminales) en los subdiagramas correspondientes de la estructura Case.

Decoraciones

Use las decoraciones situadas en la paleta **Decorations** para agrupar o separar objetos del panel frontal con cuadros, líneas o flechas. Los objetos de la paleta **Decorations** no muestran datos.

Menús

Use menús personalizados para presentar la funcionalidad del panel frontal ordenadamente y con relativamente poco espacio. Si usa poco espacio, dejará hueco en el panel frontal para controles e indicadores importantes, elementos para principiantes, elementos necesarios para productividad y elementos que no encajen bien en los menús. También puede crear métodos abreviados de teclado para elementos de menú.

Para crear un menú contextual en tiempo de ejecución para objetos del panel frontal, haga clic con el botón derecho en el objeto del panel frontal y seleccione **Advanced»Run-Time Shortcut Menu»Edit**. Para crear un menú personalizado en tiempo de ejecución para su VI, seleccione **Edit» Run-Time Menu**.

Cambio de tamaño automático para objetos del panel frontal

Use las opciones **File»VI Properties»Window Size** para configurar el tamaño mínimo de una ventana, mantener la proporción de ésta durante los cambios de la pantalla y configurar objetos del panel frontal para cambiar su tamaño de dos modos diferentes. Cuando diseñe un VI, piense en si el panel frontal puede verse en ordenadores con distintas resoluciones de pantalla. Seleccione **File»VI Properties**, después **Window Size** en el menú desplegable **Category** y active la casilla de verificación **Maintain proportions of window for different monitor resolutions** para mantener las proporciones de la ventana del panel frontal relativas a la resolución de la pantalla.

B. Tipos de datos de LabVIEW

Existen muchos tipos de datos. Ya aprendió tipos de datos numéricos, booleanos y de cadena de caracteres en la lección 2, *Explorando LabVIEW*. Otros tipos de datos son los enumerados, los dinámicos y otros. Incluso dentro de los tipos de datos numéricos se distingue entre números enteros y fraccionales.

Terminales

Los terminales del diagrama de bloques comunican visualmente al usuario información sobre el tipo de datos que representan. Por ejemplo, en la figura 4-4, **Altura (cm)** es un dato numérico en coma flotante y doble precisión. Esto se indica por el color del terminal, naranja, y por el texto del terminal, DBL.

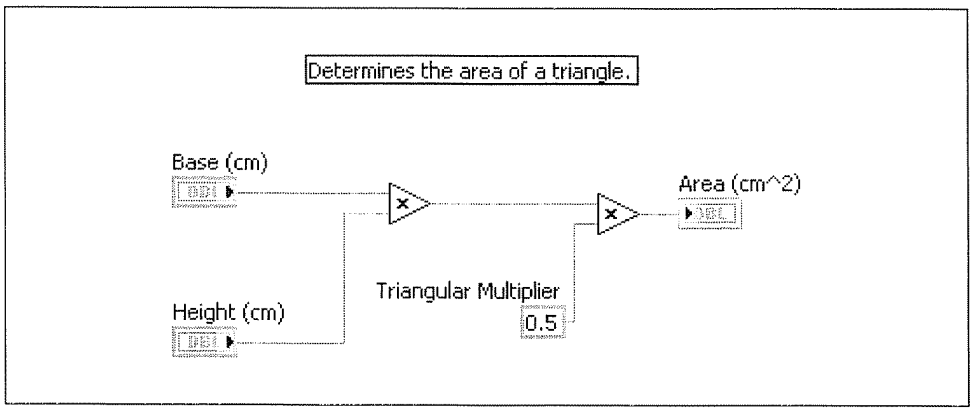


Figura 4-4. Ejemplo de tipo de datos del terminal

Tipos de datos numéricos

El tipo de dato numérico representa números de varias clases. Para cambiar el tipo de representación de un número, haga clic con el botón derecho en el control, indicador o constante y seleccione **Representation**, como se ve en la figura 4-5.

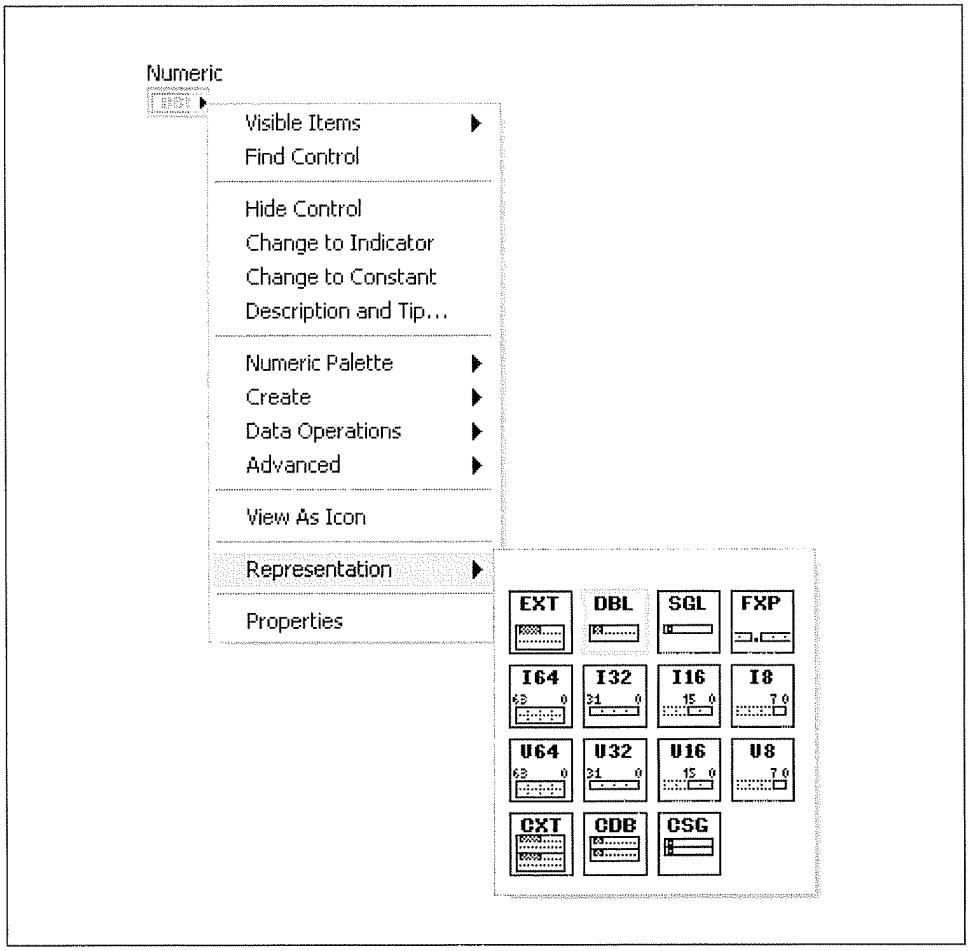


Figura 4-5. Representación numérica

Cuando cablea dos o más entradas numéricas de distintas representaciones a una función, ésta normalmente devuelve los datos en la representación más grande o amplia. Las funciones convierten las representaciones más pequeñas a la representación más amplia antes de la ejecución y LabVIEW coloca un punto de conversión en el terminal donde se realiza la conversión.

El tipo de dato numérico incluye estas subcategorías de representación: números en coma flotante, números enteros con signo, números enteros sin signo y números complejos.

Números en coma flotante

Los números en coma flotante representan números fraccionales. En LabVIEW, los números en coma flotante se representan con el color naranja.

Precisión simple (SGL): los números en coma flotante de precisión simple tienen un formato de precisión simple de 32 bits IEEE. Use los números en coma flotante de precisión simple para ahorrar memoria y evitar el desbordamiento del rango de los números.

Doble precisión (DBL): los números en coma flotante de doble precisión tienen un formato de precisión doble de 64 bits IEEE. La doble precisión es el formato predeterminado para objetos numéricos. Use los números en coma flotante de doble precisión para la mayoría de las situaciones.

Precisión ampliada (EXT): en la memoria, el tamaño y la precisión de números de precisión ampliada varían en función de la plataforma. En Windows tienen un formato de precisión ampliada de 80 bits IEEE.

Tipo de datos de punto fijo

El tipo de datos de punto fijo es numérico y representa un conjunto de números racionales que utilizan dígitos binarios o bits. A diferencia del tipo de datos en coma flotante, que permite que varíe el número total de bits que usa LabVIEW para representar números, puede configurar los números de punto fijo para que usen siempre un número concreto de bits. Así, el hardware y los destinos que sólo pueden almacenar y procesar datos con un número limitado o fijo de bits podrán almacenar y procesar los números. Puede especificar el rango y la precisión de números de punto fijo.

Nota Para representar un número racional con el tipo de datos de punto fijo, el denominador del número racional debe ser una potencia de 2, porque el sistema de números binarios es de base 2.

Use el tipo de datos de punto fijo cuando no necesite la funcionalidad dinámica de la representación de coma flotante o cuando desee trabajar con un destino que no admita la aritmética de coma flotante, como un destino FPGA.

Especifique la codificación, la longitud de palabra y la longitud de palabra entera de un número de punto fijo cuando desee que el número se adapte a cierto tamaño de bits.

Codificación: la codificación binaria del número de punto fijo. Puede seleccionar con signo o sin signo. Si selecciona con signo, el bit del signo es siempre el primero en la cadena de caracteres de bits que representa los datos.

Longitud de palabra: el número total de bits de la cadena de caracteres de bits que usa LabVIEW para representar todos los valores posibles de los datos de punto fijo. LabVIEW acepta una longitud máxima de palabra de 64 bits. Ciertos destinos pueden limitar los datos a longitudes menores de palabra. Si abre un VI en un destino y el VI contiene datos de punto fijo con longitudes de palabra mayores que las que puede aceptar el destino, el VI contendrá cables rotos. Consulte la documentación de un destino para determinar la máxima longitud de palabra que acepta.

Longitud de palabra entera: el número de bits enteros en una cadena de caracteres de bits que usa LabVIEW para representar todos los posibles valores del dato de punto fijo o, dada una posición inicial a la izquierda o a la derecha del bit más significativo, el número de bits a desplazar la coma binaria y llegar al bit más significativo. La longitud de palabra del número entero puede ser mayor que la longitud de palabra, y puede ser positivo o negativo.

Números enteros

Los números enteros representan los números naturales. Los números enteros con signo pueden ser positivos o negativos. Use los tipos de datos de números enteros sin signo cuando sepa que el número entero es siempre positivo. En LabVIEW, los números enteros se representan con el color azul.

Cuando LabVIEW convierte los números en coma flotante en números enteros, el VI redondea al número entero par más cercano. Por ejemplo, LabVIEW redondea 2,5 a 2 y 3,5 a 4.

Byte (I8): los números enteros Byte tienen 8 bits de almacenamiento.

Word (I16): los números enteros Word tienen 16 bits de almacenamiento.

Long (I32): los números enteros Long tienen 32 bits de almacenamiento. En la mayoría de los casos es mejor utilizar un número entero de 32 bits.

Quad (I64): los números enteros Quad tienen 64 bits de almacenamiento.

Números complejos

Los números complejos se representan con dos valores unidos en la memoria: uno representa la parte real y otro la parte imaginaria. En LabVIEW, como los números complejos son un tipo de número en coma flotante, los números complejos también se representan con el color naranja.

Complejo simple (CSG): los números complejos de coma flotante de precisión simple constan de valores reales e imaginarios y tienen un formato de precisión simple de 32 bits IEEE.

Complejo doble (CDB): los números complejos de coma flotante de doble precisión constan de valores reales e imaginarios y tienen un formato de doble precisión de 64 bits IEEE.

Complejo ampliado (CXT): los números complejos de coma flotante de precisión ampliada constan de valores reales e imaginarios y tienen un formato de precisión ampliada IEEE. En la memoria, el tamaño y la precisión de números de precisión ampliada varían en función de la plataforma. En Windows tienen un formato de precisión ampliada de 80 bits IEEE.

Valores booleanos

LabVIEW almacena datos booleanos como valores de 8 bits. Si el valor de 8 bits es cero, el valor booleano es FALSE. Cualquier valor que no sea cero representará TRUE. En LabVIEW, el color verde representa datos booleanos.

Los valores booleanos también tienen una acción mecánica asociada a ellos. Las dos acciones principales son bloquear y conmutar. Puede seleccionar uno de los siguientes comportamientos de control:

- **Switch when pressed**: cambia el valor del control cada vez que hace clic en él con la herramienta de operaciones. La frecuencia con la que el VI lee el control no afecta a este comportamiento.
- **Switch when released**: cambia el valor del control sólo después de soltar el botón del ratón mientras hace clic dentro del límite gráfico del control. La frecuencia con la que el VI lee el control no afecta a este comportamiento.
- **Switch until released**: cambia el valor del control cuando hace clic en él y conserva el nuevo valor hasta que suelte el botón del ratón. En este momento, el control vuelve a su valor predeterminado, al igual que el funcionamiento del timbre de una puerta. La frecuencia con la que el VI lee el control no afecta a este comportamiento. No puede seleccionar este comportamiento para un control de botones de radio.
- **Latch when pressed**: cambia el valor del control cuando hace clic en él y conserva el nuevo valor hasta que el VI lo lea una vez. En este momento, el control vuelve a su valor predeterminado, aunque siga pulsando el botón del ratón. Este comportamiento es similar al de un disyuntor y es útil para detener un bucle While o para que el VI realice una acción sólo una vez cada vez que ajuste el control. No puede seleccionar este comportamiento para un control de botones de radio.

- **Latch when released**: cambia el valor del control sólo después de soltar el botón del ratón dentro del límite gráfico del control. Cuando el VI lo lee una vez, el control vuelve a su valor predeterminado. Este comportamiento funciona del mismo modo que los botones de los cuadros de diálogo y del sistema. No puede seleccionar este comportamiento para un control de botones de radio.
- **Latch until released**: cambia el valor del control cuando hace clic en él y conserva el valor hasta que el VI lo lea una vez o suelte el botón del ratón, en función de lo que ocurra en último lugar. No puede seleccionar este comportamiento para un control de botones de radio.

Para saber más de la acción mecánica, experimente con el VI Mechanical Action of Booleans en el NI Example Finder.

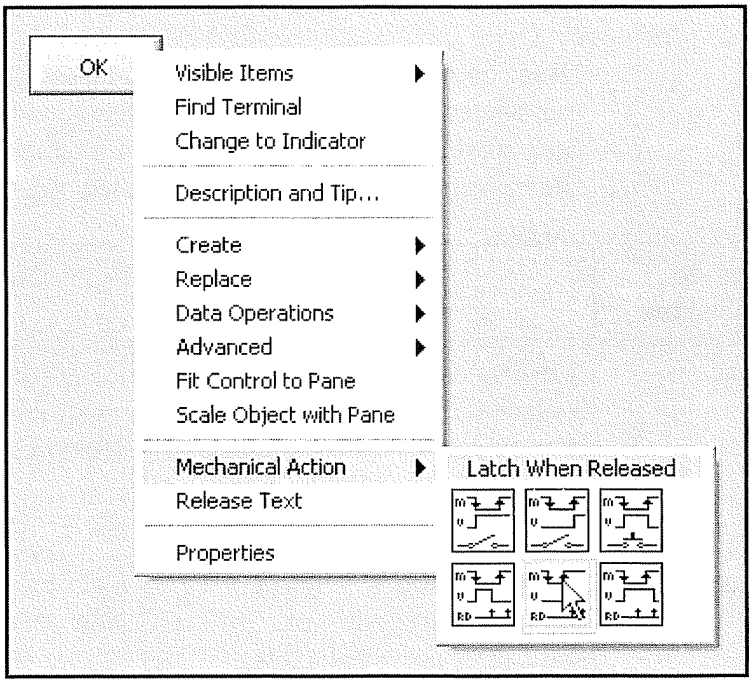


Figura 4-6. Acción mecánica booleana

Cadenas de caracteres

Una cadena de caracteres es una secuencia de caracteres ASCII visualizables y no visualizables. Las cadenas de caracteres tienen un formato independiente de las plataformas para la información y los datos. Algunas de las aplicaciones más comunes de las cadenas de caracteres son:

- Generación de mensajes de texto simples
- Control de instrumentos enviando comandos de texto al instrumento y devolviendo valores de datos en forma de cadenas de caracteres ASCII o binarias que después convierte en valores numéricos.

- Almacenamiento de datos numéricos en disco. Para almacenar datos numéricos en un fichero ASCII, primero debe convertir los datos numéricos en cadenas de caracteres antes de escribir los datos en un fichero del disco.
- Petición o visualización de datos con cuadros de diálogo

En el panel frontal, las cadenas aparecen como tablas, cuadros de entrada de texto y etiquetas. LabVIEW incluye VIs y funciones integradas que puede utilizar para manipular cadenas de caracteres, como dar formato a cadenas de caracteres, analizar cadenas de caracteres y otras modificaciones.

Consulte el tema *ASCII Codes* de la *Ayuda de LabVIEW* para obtener información acerca de códigos ASCII y funciones de conversión.

En LabVIEW, las cadenas de caracteres se representan con el color rosa.

Haga clic con el botón derecho en el control o indicador de una cadena de caracteres del panel frontal para seleccionar entre los tipos de visualización que aparecen en la siguiente tabla. La tabla también muestra un ejemplo de mensaje para cada tipo de visualización.

Tipo de visualización	Descripción	Mensaje
Visualización normal	Muestra caracteres imprimibles utilizando la fuente del control. Los caracteres no visualizables normalmente aparecen como cuadros.	`Existen cuatro tipos de visualización.\ es una barra invertida.`
Visualización de códigos '\'	Muestra códigos de barra invertida para todos los caracteres no visualizables.	`Existen\scuatro\stipos\ sde\svisualización.\n\\\ses\ suna\sbarra\sinvertida.`
Visualización de contraseña	Muestra un asterisco (*) para cada carácter, incluidos los espacios.	`**********`
Visualización hexadecimal	Muestra el valor ASCII de cada carácter en hexadecimal en lugar del carácter.	`5468 6572 6520 6172 6520 666F 7572 2064 6973 706C 6179 2074 7970 6573 2E0A 5C20 6973 2061 2062 6163 6B73 6C61 7368 2E`

LabVIEW almacena cadenas de caracteres como un puntero a una estructura que contiene un valor con una longitud de 4 bytes seguido de un array 1D de números enteros de bytes (caracteres de 8 bits).

Enums

Un enum (indicador, constante o control enumerado) es una combinación de tipos de datos. Un enum representa una pareja de valores, una cadena de caracteres y un numérico, donde el enum puede ser uno de una lista de valores. Por ejemplo, si creó un tipo de enum llamado Mes, las parejas de valores posibles para una variable Mes son Enero-0, Febrero-1, etc. hasta Diciembre-11. La figura 4-7 muestra un ejemplo de estas parejas de datos en el cuadro de diálogo **Properties** para un control enumerado.

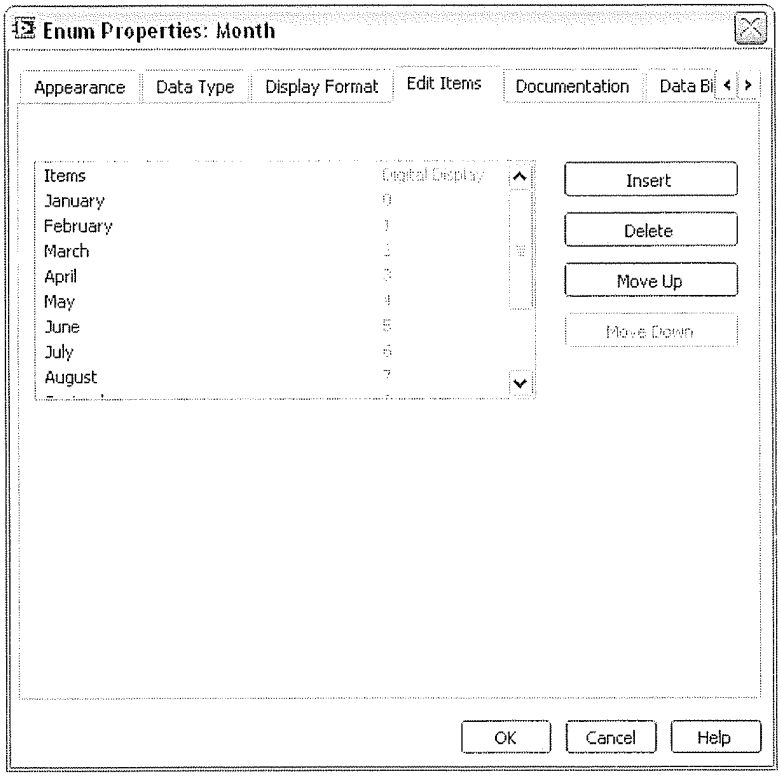
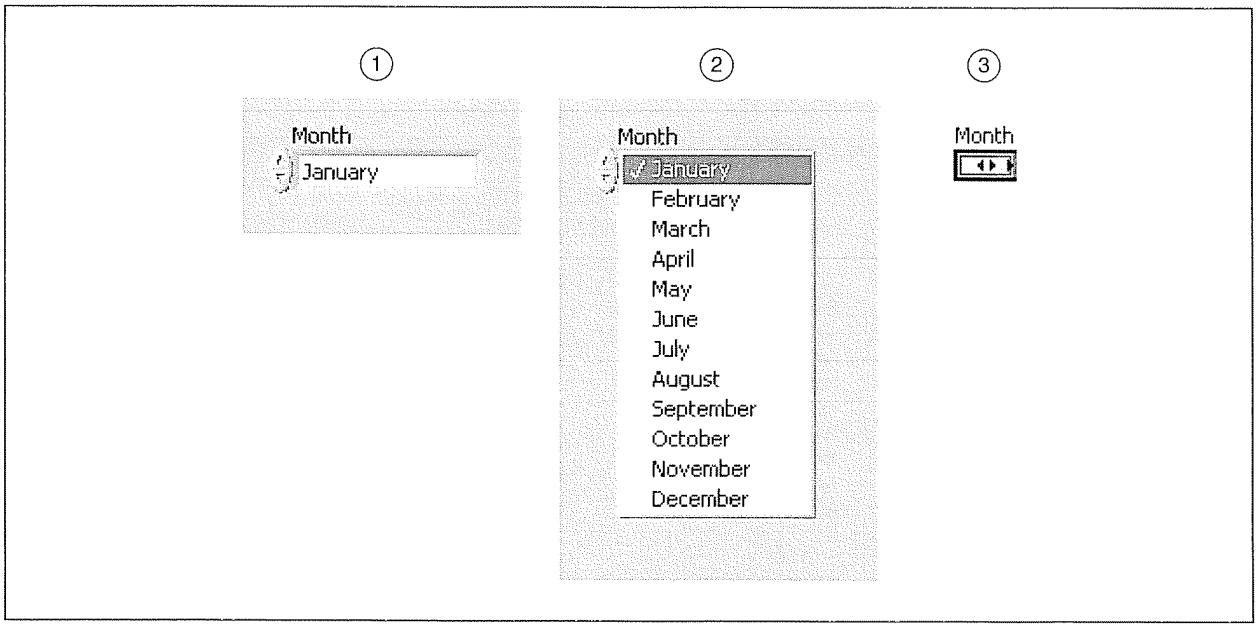


Figura 4-7. Propiedades para el control enumerado Mes

Los enums resultan útiles porque es más fácil manipular números en el diagrama de bloques que cadenas de caracteres. La figura 4-8 muestra el control enumerado **Mes (Month)**, la selección de una pareja de datos en el control enumerado y el terminal correspondiente del diagrama de bloques.



1	Control del panel frontal	3	Terminal del diagrama de bloques
2	Selección de un elemento		

Figura 4-8. Control enumerado Mes (Month)

Dinámicos

El tipo de datos dinámico almacena la información que genera o adquiere un VI Express. El tipo de datos dinámico aparece como un terminal azul oscuro, mostrado a la izquierda. La mayoría de VIs Express aceptan o devuelven el tipo de datos dinámico. Puede cablear el tipo de datos dinámico a cualquier indicador o entrada que acepte datos numéricos, de forma de onda o booleanos. Cablee el tipo de datos dinámico a un indicador que pueda presentar mejor los datos. Ejemplos de indicadores son gráficos tipo "graph", gráficos tipo "chart" o indicadores numéricos.

La mayoría del resto de VIs y funciones de LabVIEW no aceptan el tipo de datos dinámico. Para utilizar un VI o función integrada para analizar o procesar los datos que incluye el tipo de datos dinámico, debe convertir el tipo de datos dinámico.

Use el VI Express Convert from Dynamic Data para convertir el tipo de datos dinámico en tipos de datos numéricos, de forma de onda y de array para usar con otros VIs y funciones. Cuando coloca el VI Express Convert from Dynamic Data en el diagrama de bloques, aparece el cuadro de diálogo

Configure Convert from Dynamic Data. El cuadro de diálogo **Configure Convert from Dynamic Data** muestra opciones para especificar cómo desea dar formato a los datos que devuelve el VI Express Convert from Dynamic Data.

Cuando cablea un tipo de datos dinámico a un indicador de array, LabVIEW añade automáticamente el VI Express Convert from Dynamic Data al diagrama de bloques. Haga doble clic en el VI Express Convert from Dynamic Data para abrir el cuadro de diálogo **Configure Convert from Dynamic Data** y controlar cómo aparecerán los datos en el array.

C. Documentación de código

Los desarrolladores profesionales que mantienen y modifican VIs conocen el valor de una buena documentación. Documente bien el diagrama de bloques para facilitar las futuras modificaciones del código. Además, documente bien la ventana del panel frontal para explicar el objetivo del VI y los objetos del panel frontal.

Use ayudas, descripciones, propiedades de VI y un buen diseño para documentar las ventanas del panel frontal.

Ayudas y descripciones

Las ayudas son breves descripciones que aparecen cuando mueve el cursor sobre un control o indicador mientras se ejecuta un VI. Por ejemplo, podría añadir una ayuda para indicar que una temperatura es en grados Celsius o explicar cómo funciona una entrada en un algoritmo. Las descripciones ofrecen información adicional sobre controles e indicadores específicos. Las descripciones aparecen en la ventana **Context Help** cuando mueve el cursor sobre el objeto. Para añadir ayudas y descripciones a controles, haga clic con el botón derecho en el control o indicador y seleccione **Description and Tip** desde el menú contextual.

Propiedades de VI

Use el componente Documentation del cuadro de diálogo **VI Properties** para crear descripciones de VIs y enlazar desde VIs a ficheros HTML o a ficheros de ayuda compilados. Para ver VI Properties, haga clic con el botón derecho en el icono del VI en el panel frontal o en el diagrama de bloques y seleccione **VI Properties** desde el menú contextual o seleccione **File»VI Properties**. Después seleccione **Documentation** en el menú desplegable **Categories**. No puede acceder a este cuadro de diálogo mientras se ejecuta un VI.

Esta página incluye los siguientes componentes:

- **VI description**: contiene el texto que aparece en la ventana **Context Help** cuando mueve el cursor sobre el icono de VI. Use las etiquetas <B> y </B> alrededor de cualquier texto de la descripción al que desee dar formato en negrita. También puede utilizar la propiedad VI Description para modificar la descripción del VI programáticamente.
- **Help tag**: contiene el nombre del fichero HTML o la palabra clave del índice del tema que desea vincular en un fichero de ayuda compilado. También puede utilizar la propiedad Help:Document Tag para configurar la etiqueta de ayuda programáticamente.
- **Help path**: contiene la ruta al fichero HTML o al fichero de ayuda compilado que desea vincular desde la ventana Context Help. Si este campo está vacío, el enlace **Detailed help** no aparecerá en la ventana Context Help y se atenuará el botón **Detailed help**.
- **Browse**: muestra un cuadro de diálogo de fichero que se utiliza para navegar hasta un fichero HTML o hasta un fichero de ayuda compilado para utilizarlo como ruta de ayuda.

Nombre de controles e indicadores

Si da nombres lógicos y descriptivos a controles e indicadores, facilitará el uso de los paneles frontales. Por ejemplo, si llama a un control `Temperature`, un usuario quizá no sepa qué unidades utilizar. Sin embargo, si lo llama `Temperature °C`, añadirá más información al panel frontal. Ahora sabrá que debe introducir temperaturas en unidades métricas.

Programación gráfica

Aunque la naturaleza gráfica de LabVIEW ayuda a la autodocumentación de diagramas de bloques, los comentarios adicionales ayudan a la hora de modificar sus VIs posteriormente. Existen dos tipos de comentarios de diagrama de bloques: los que describen la función u operación de algoritmos y comentarios que explican la finalidad de datos que pasan por cables. Ambos tipos de comentarios aparecen en el siguiente diagrama de bloques. Puede insertar etiquetas estándares con la herramienta de etiquetado, o insertando una etiqueta libre desde la subpaleta **Functions»Programming» Structures»Decorations**. De forma predeterminada las etiquetas libres tienen un color de fondo amarillo.

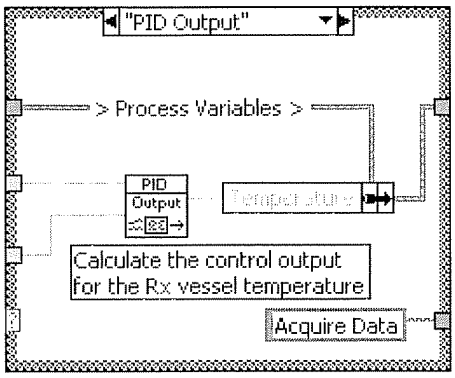


Figura 4-9. Documentar un diagrama de bloques

Use las siguientes pautas para comentar los VIs:

- Use comentarios en el diagrama de bloques para explicar lo que está realizando el código.
- Aunque el código de LabVIEW puede autodocumentarse porque es gráfico, use etiquetas libres para describir cómo funciona el diagrama de bloques.
- No muestre etiquetas en llamadas de función y de subVIs, ya que suelen ser grandes y poco manejables. Un desarrollador que observe el diagrama de bloques puede encontrar el nombre de una función o subVI en la ventana **Context Help**.
- Use etiquetas libres pequeñas con fondos blancos para etiquetar cables largos e identificar su uso. Etiquetar cables resulta útil para los que proceden de registros de desplazamiento y para cables largos que pueden abarcar todo el diagrama de bloques. Consulte la sección *Estructuras Case* de esta lección para obtener información adicional acerca de los registros de desplazamiento.
- Etiquete estructuras para especificar su funcionalidad principal.
- Etiquete constantes para especificar su naturaleza.
- Use etiquetas libres para documentar algoritmos que usa en los diagramas de bloques. Si usa un algoritmo de un libro u otra referencia, incluya la información de referencia.

Ejercicio 4-1 VI Determine Warnings

Objetivo

Crear y documentar un VI sencillo.

Caso

Debe crear una parte de un proyecto mayor. El desarrollador principal le proporciona las entradas del VI, el algoritmo y las posibles salidas. Cree y documente un VI en función del diseño dado.

Diseño

Entradas y salidas

Tipo	Nombre	Propiedades
Control numérico	`Current Temp`	Doble precisión y coma flotante
Control numérico	`Max Temp`	Doble precisión y coma flotante
Control numérico	`Min Temp`	Doble precisión y coma flotante
Indicador de cadena de caracteres	`Warning Text`	Tres valores en potencia: advertencia de golpe de calor, sin advertencia y advertencia de congelación
Indicador LED redondo	`Warning?`	—

Diagrama de flujo

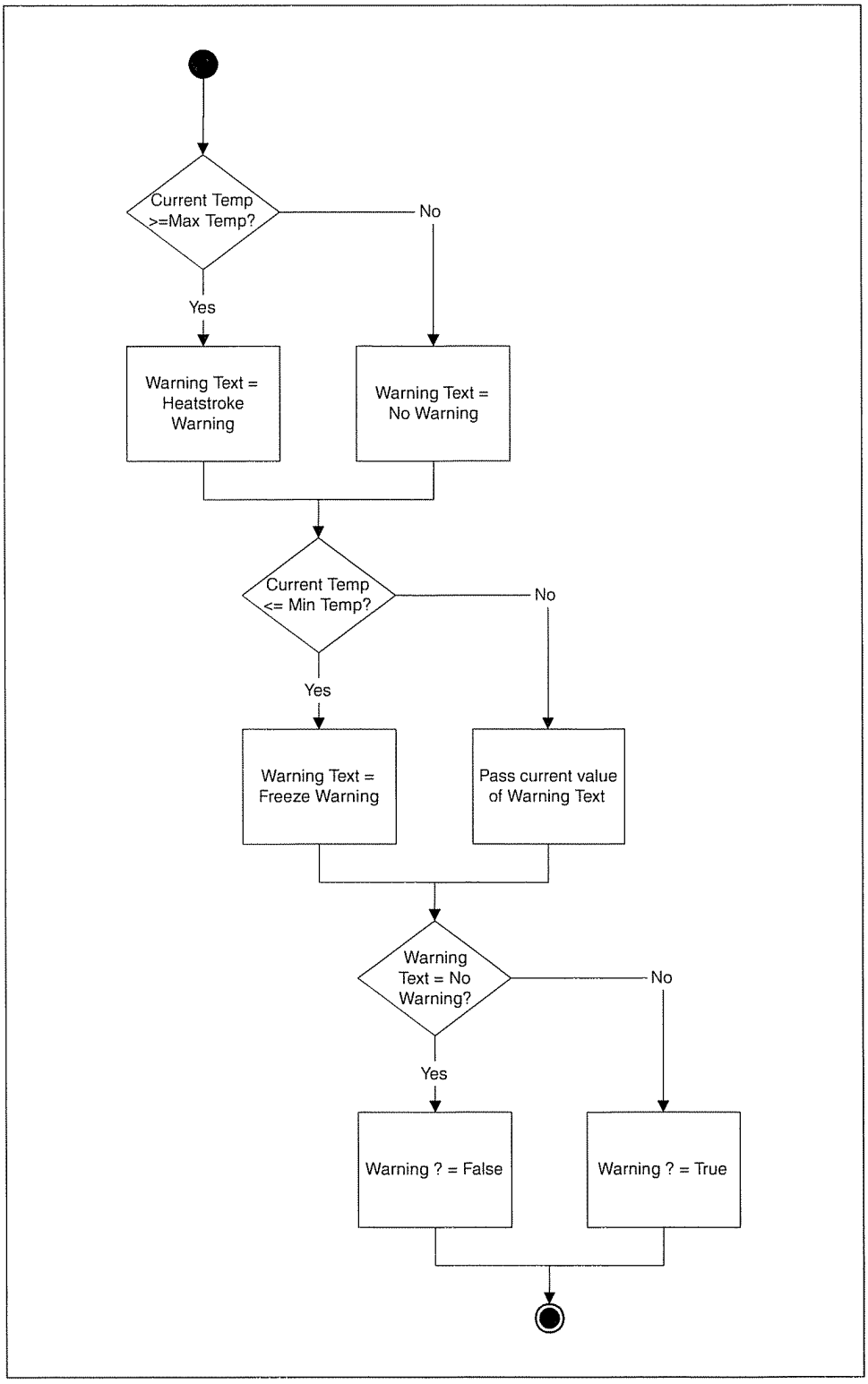


Figura 4-10. Diagrama de flujo del VI Determine Warnings

Implementación

Siga las instrucciones inferiores para crear un panel frontal similar al de la figura 4-11. El usuario introduce la temperatura actual, la máxima y la mínima. A continuación, el panel frontal muestra la cadena de caracteres de advertencia y el LED booleano de advertencia. Este VI forma parte del proyecto de la estación meteorológica de temperatura estudiado durante el curso.

1. Abra un VI nuevo y cree el siguiente panel frontal.

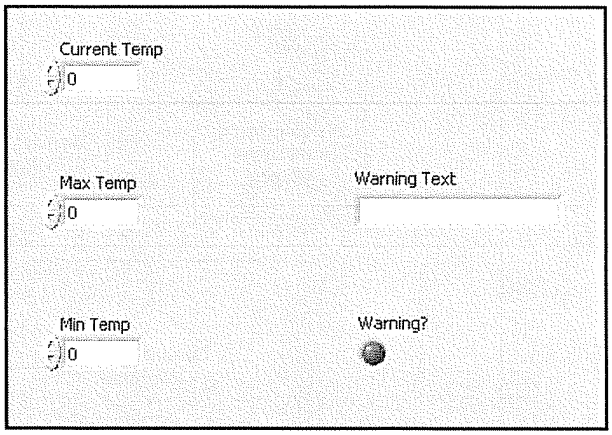


Figura 4-11. Panel frontal del VI Determine Warnings

2. Guarde el nuevo VI.

   - ❑ Seleccione **File»Save**.
   - ❑ Guarde el VI como `Determine Warnings.vi` en el directorio `<Exercises>\LabVIEW Basics I\Determine Warnings`.

3. Cree un control numérico para la temperatura actual.

   - ❑ Añada un control numérico a la ventana del panel frontal.
   - ❑ Cambie la etiqueta del control numérico a `Current Temp`.
   - ❑ Haga clic con el botón derecho en el control, seleccione **Representation** y confirme que el tipo de representación está configurado en doble precisión.

Consejo Este subVI podría utilizarse para Fahrenheit, Kelvin u otra escala de temperatura, siempre que todas las entradas utilicen la misma escala. Por lo tanto, no es necesario añadir unidades de escala a las etiquetas.

4. Cree un control numérico para la temperatura máxima.

   - ❑ Mantenga pulsada la tecla <Ctrl> y haga clic y arrastre el control numérico Current Temp para crear una copia del control.
   - ❑ Cambie el texto de la etiqueta del nuevo control numérico a `Max Temp`.

5. Cree un control numérico para la temperatura mínima.

   - ❑ Mantenga pulsada la tecla <Ctrl> y haga clic y arrastre el control numérico Max Temp para crear una copia del control.
   - ❑ Cambie el texto de la etiqueta del nuevo control numérico a `Min Temp`.

6. Cree un indicador de cadena de caracteres para el texto de advertencia.

   - ❑ Añada un indicador de cadena de caracteres a la ventana del panel frontal.
   - ❑ Cambie el texto de la etiqueta del indicador de cadena de caracteres a `Warning Text`.

7. Cree un LED redondo u otro indicador booleano para el booleano de advertencia.

   - ❑ Añada un LED redondo a la ventana del panel frontal.
   - ❑ Cambie el texto de la etiqueta del indicador booleano a `Warning?`.

8. Cambie al diagrama de bloques.

Consejo Si no desea ver terminales como iconos en el diagrama de bloques, seleccione **Tools»Options** y después **Block Diagram** de la lista **Category**. Elimine la marca del elemento **Place front panel terminals as icons**.

Siga las instrucciones inferiores para crear un diagrama de bloques similar al de la figura 4-12.

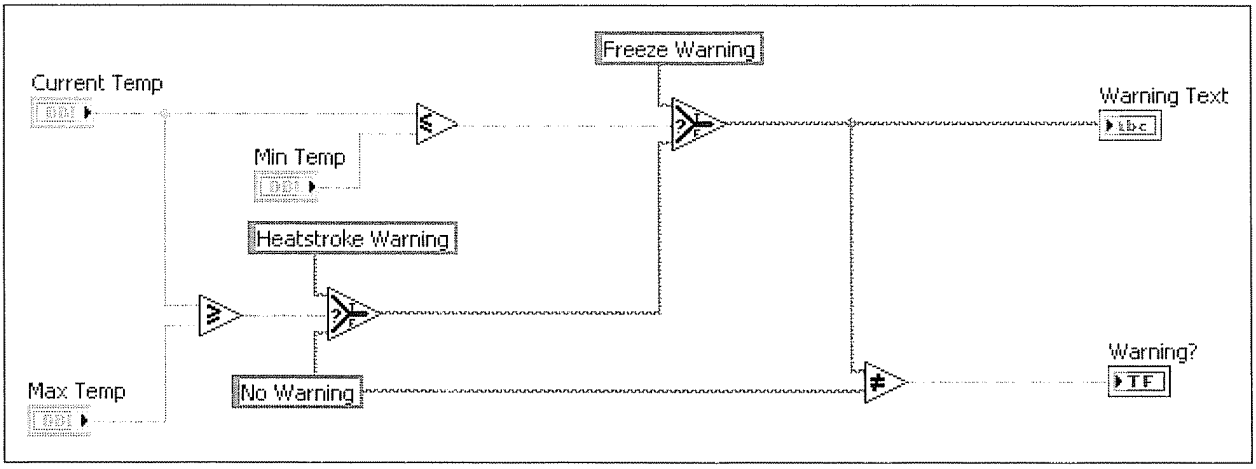


Figura 4-12. Diagrama de bloques del VI Determine Warnings

9. Compare `Current Temp` y `Max Temp`.

   ❑ Añada una función Greater Or Equal? al diagrama de bloques.

   ❑ Cablee el control **Current Temp** a la entrada **x** de la función Greater Or Equal?.

   ❑ Cablee el control **Max Temp** a la entrada **y** de la función Greater Or Equal?.

10. Compare `Current Temp` y `Min Temp`.

    ❑ Añada una función Less Or Equal? al diagrama de bloques.

    ❑ Cablee el control **Current Temp** a la entrada **x** de la función Less Or Equal?.

    ❑ Cablee el control **Min Temp** a la entrada **y** de la función Less Or Equal?.

11. Si el valor **Current Temp** es igual o mayor que **Max Temp**, genere una cadena de caracteres `Heatstroke Warning`, de lo contrario genere una cadena de caracteres `No Warning`.

    ❑ Añada la función Select al diagrama de bloques a la derecha de la función Greater Or Equal?.

    ❑ Cablee la salida de la función Greater Or Equal? a la entrada **s** de la función Select.

- ❑ Añada una constante de cadena de caracteres al diagrama de bloques a la parte superior izquierda de la función Select.
- ❑ Escriba `Heatstroke Warning` en la constante de cadena de caracteres.
- ❑ Cablee la cadena de caracteres `Heatstroke Warning` a la entrada **t** de la función Select.
- ❑ Mantenga pulsada la tecla <Ctrl> y haga clic y arrastre la constante de cadena de caracteres `Heatstroke Warning` a la parte inferior izquierda de la función Select para crear una copia de la constante.
- ❑ Escriba `No Warning` en la segunda constante de cadena de caracteres.
- ❑ Cablee la cadena de caracteres `No Warning` a la entrada **f** de la función Select.

12. Si la **Current Temp** es igual o menor que la **Min Temp**, genere una cadena de caracteres `Freeze Warning`, de lo contrario use la cadena de caracteres generada en el paso 11.
    - ❑ Cree una copia de la función Select y colóquela a la derecha de la función Less Or Equal?.
    - ❑ Cablee la salida de la función Less Or Equal? a la entrada **s** de la función Select.
    - ❑ Cree una copia de la constante de cadena de caracteres y colóquela en la parte superior derecha de la función Select.
    - ❑ Escriba `Freeze Warning` en la constante de cadena de caracteres.
    - ❑ Cablee la cadena de caracteres `Freeze Warning` a la entrada **t** de la función Select.
    - ❑ Cablee la salida de la función Select anterior a la entrada **f** de la nueva función Select.

13. Muestre el texto generado.

    - ❑ Cablee la salida de la segunda función Select con el indicador Warning Text.

14. Genere el control booleano **Warning?** determinando si el valor de **Warning Text** es igual a `No Warning`.

    - ❑ Añada una función Not Equal? a la izquierda del control booleano **Warning?**.

    - ❑ Cablee la salida de la segunda función Select a la entrada **x** de la función Not Equal?.

    - ❑ Cablee la constante de cadena de caracteres `No Warning` a la entrada **y** de la función Not Equal?.

    - ❑ Cablee la salida de la función Not Equal? al control Warning?.

15. Documente el código utilizando estas sugerencias en el panel frontal.

    - ❑ Cree ayudas para cada control e indicador que describa la finalidad y las unidades del objeto. Para acceder a las ayudas, haga clic con el botón derecho en un objeto y seleccione **Description and Tip**.

    - ❑ Documente las propiedades del VI, ofreciendo una descripción general del VI, una lista de entradas y salidas, su nombre y la fecha en que se creó el VI. Para acceder al cuadro de diálogo **VI Properties**, seleccione **File»VI Properties**.

    - ❑ Documente el algoritmo del diagrama de bloques con una etiqueta libre.

16. Guarde el VI.

Prueba

1. Pruebe el VI introduciendo un valor para **Current Temp**, **Max Temp** y **Min Temp** y ejecutando el VI para cada configuración.

   La tabla 4-1 muestra la cadena de caracteres esperada **Warning Text** y el valor booleano **Warning?** para cada ajuste de valores de entrada.

Tabla 4-1. Probar valores para el VI Determine Warnings

Current Temp	Max Temp	Min Temp	Warning Text	Warning?
30	30	10	`Heatstroke Warning`	True
25	30	10	`No Warning`	False
10	30	10	`Freeze Warning`	True

   ¿Qué ocurre si introduce un valor **Max Temp** inferior a **Min Temp**? ¿Qué cree que sucedería? Aprenderá a manejar problemas similares en el ejercicio 4-6.

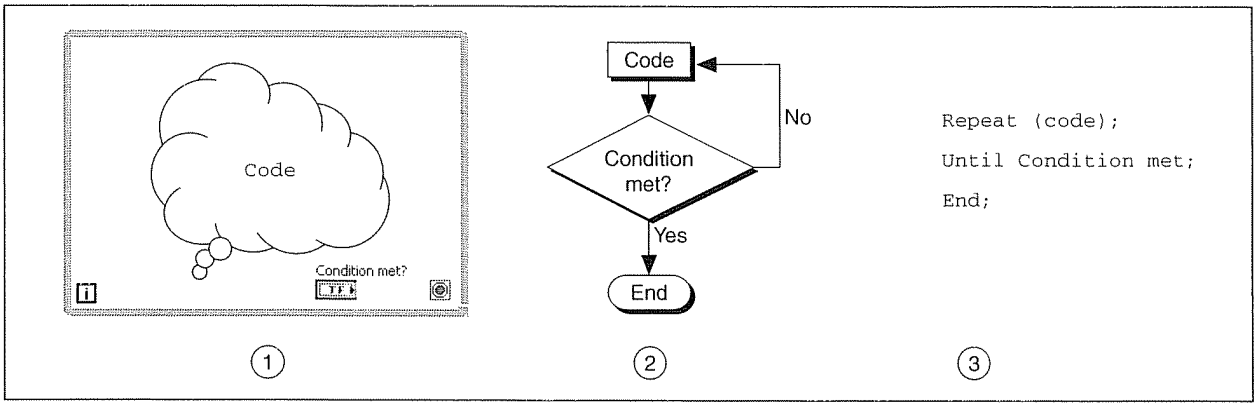
2. Guarde y cierre el VI.

Fin del ejercicio 4-1

D. Bucles While

Al igual que un bucle Do o un bucle Repeat-Until en lenguajes de programación basados en texto, un bucle While, mostrado en la figura 4-13, ejecuta un subdiagrama hasta que ocurra una condición.

La siguiente ilustración muestra un bucle While en LabVIEW, un diagrama de flujo equivalente a la funcionalidad del bucle While, y un ejemplo de pseudocódigo de la funcionalidad del bucle While.



1 Bucle While de LabVIEW	2 Diagrama de flujo	3 Pseudocódigo

Figura 4-13. Bucle While

El bucle While se encuentra en la paleta **Structures**. Seleccione el bucle While en la paleta y después use el cursor para arrastrar un rectángulo de selección alrededor de la sección del diagrama de bloques que desee repetir. Cuando suelte el botón del ratón, un límite de bucle While encerrará la sección que seleccionó.

Añada objetos del diagrama de bloques al bucle While arrastrándolos y soltándolos dentro de éste.

Consejo El bucle While siempre se ejecuta al menos una vez.

El bucle While ejecuta el subdiagrama hasta que el terminal condicional, un terminal de entrada, reciba un valor booleano específico. El terminal condicional en un bucle While actúa del mismo modo que en un bucle For con un terminal condicional. No obstante, como el bucle For también incluye un contaje de iteraciones configurado, no se ejecuta hasta el infinito si la condición nunca ocurre. El bucle While no incluye un contaje de iteraciones configurado y se ejecuta hasta el infinito si la condición nunca ocurre.

Si un terminal condicional es **Stop if True**, coloca el terminal de un control booleano fuera de un bucle While, y si el control es FALSE cuando se inicia el bucle, provocará un bucle infinito, como se muestra en el siguiente ejemplo. También provoca un bucle infinito si el terminal condicional es **Continue if True** y el control fuera del bucle se configura en TRUE.

Para convertir un bucle While en uno For, haga clic con el botón derecho en el bucle While y seleccione **Replace with For Loop** en el menú contextual. **(Windows, ETS)** Para convertir un bucle While en uno Timed, haga clic con el botón derecho en el bucle While y seleccione **Replace with Timed Loop** en el menú contextual.

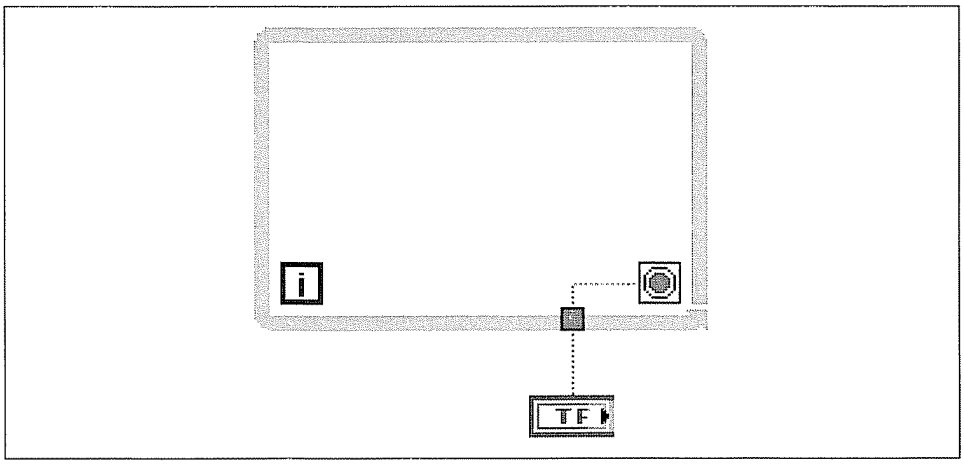


Figura 4-14. Bucle infinito

Si cambia el valor del control, no detendrá el bucle infinito, porque el valor sólo se lee una vez, antes de que empiece el bucle. Para detener un bucle infinito, debe abortar el VI haciendo clic en el botón **Abort Execution** en la barra de herramientas.

También puede realizar un control de errores básicos con el terminal condicional de un bucle While. Cuando cablea un cluster de error al terminal condicional, sólo pasa al terminal el valor TRUE o FALSE del parámetro **status** del cluster de error. Asimismo, los elementos del menú contextual **Stop if True** y **Continue if True** cambian a **Stop if Error** y **Continue while Error**.

Consulte la sección *Gestión de errores* del capítulo 6, *Ejecución y depuración de VIs*, para obtener información adicional acerca de clusters de errores y control de errores.

El terminal de iteración, que se ve abajo, es un terminal de salida que contiene el numero de iteraciones completadas.

El contaje de iteraciones siempre empieza en cero. Durante la primera iteración, el terminal de iteración devuelve `0`.

En el siguiente diagrama de bloques, el bucle While se ejecuta hasta que la salida de la función Random Number sea mayor o igual que `10,00` y el control **Enable** sea `True`. La función And devuelve `True` sólo si ambas entradas son `True`. De lo contrario, devolverá `False`.

En el siguiente ejemplo, existe mayor posibilidad de un bucle infinito. Generalmente, el comportamiento deseado es que se cumpla una condición para detener el bucle, en lugar de necesitar que se cumplan ambas condiciones.

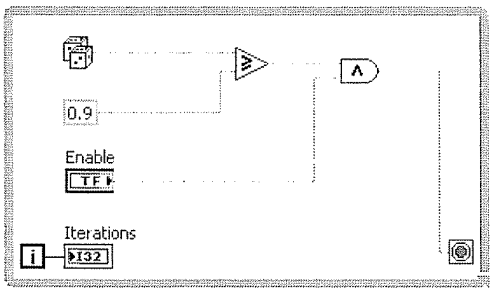


Figura 4-15. Posible bucle infinito

Túneles de estructuras

Los túneles suministran datos hacia dentro y hacia fuera de las estructuras. El túnel aparece como un bloque sólido en el borde del bucle While. El bloque es del color del tipo de datos cableado al túnel. Los datos salen de un bucle cuando éste termina. Cuando un túnel introduce datos en un bucle, éste se ejecuta sólo cuando los datos llegan al túnel.

En el siguiente diagrama de bloques, el terminal de iteración se conecta a un túnel. El valor del túnel no entra en el indicador **Iterations** hasta que el bucle While termine de ejecutarse.

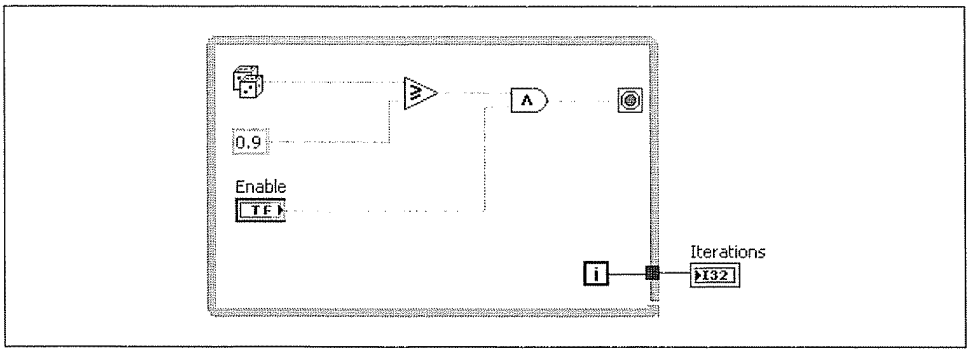


Figura 4-16. Túnel del bucle While

Sólo aparece en el indicador **Iterations** el último valor del terminal de iteración.

Uso de bucles While para gestión de errores

Puede cablear un cluster de error al terminal condicional de un bucle While o un bucle For con un terminal condicional para detener la iteración del bucle. Cuando cablea un cluster de error al terminal condicional, sólo pasa al terminal el valor TRUE o FALSE del parámetro **status** del cluster de error. Si ocurre un error, el bucle se detendrá. En un bucle For con un terminal condicional, también debe cablear un valor al terminal de contaje o autoindexar un array de entrada para configurar un número máximo de iteraciones. El bucle For se ejecuta hasta que ocurra un error o hasta que se complete el número de iteraciones configurado.

Si cablea un cluster de error al terminal condicional, los elementos del menú contextual **Stop if True** y **Continue if True** cambiarán a **Stop on Error** y **Continue while Error**.

En la figura 4-17, el cluster de error y un botón de parada se utilizan para determinar cuándo detener el bucle. Éste es el método recomendado para detener la mayoría de los bucles.

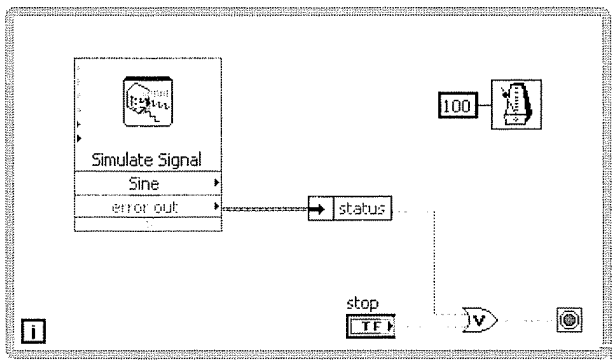


Figura 4-17. Detener un bucle While

Ejercicio 4-2 VI Auto Match

Objetivo

Usar un bucle While y un terminal de iteración y pasar datos por un túnel.

Caso

Cree un VI que genere continuamente números aleatorios entre 0 y 1.000 hasta que genere un número que coincida con el que seleccione el usuario. Determine cuántos números aleatorios generó el VI antes de que el número coincidiera.

Diseño

Tabla 4-2. Entradas y salidas

Tipo	Nombre	Propiedades
Entrada	`Number to Match`	Doble precisión, coma flotante entre 0 y 1000, convertir al número natural más cercano, valor predeterminado = 50
Salida	`Current Number`	Doble precisión y coma flotante
Salida	`Number of Iterations`	Número entero

Diagrama de flujo

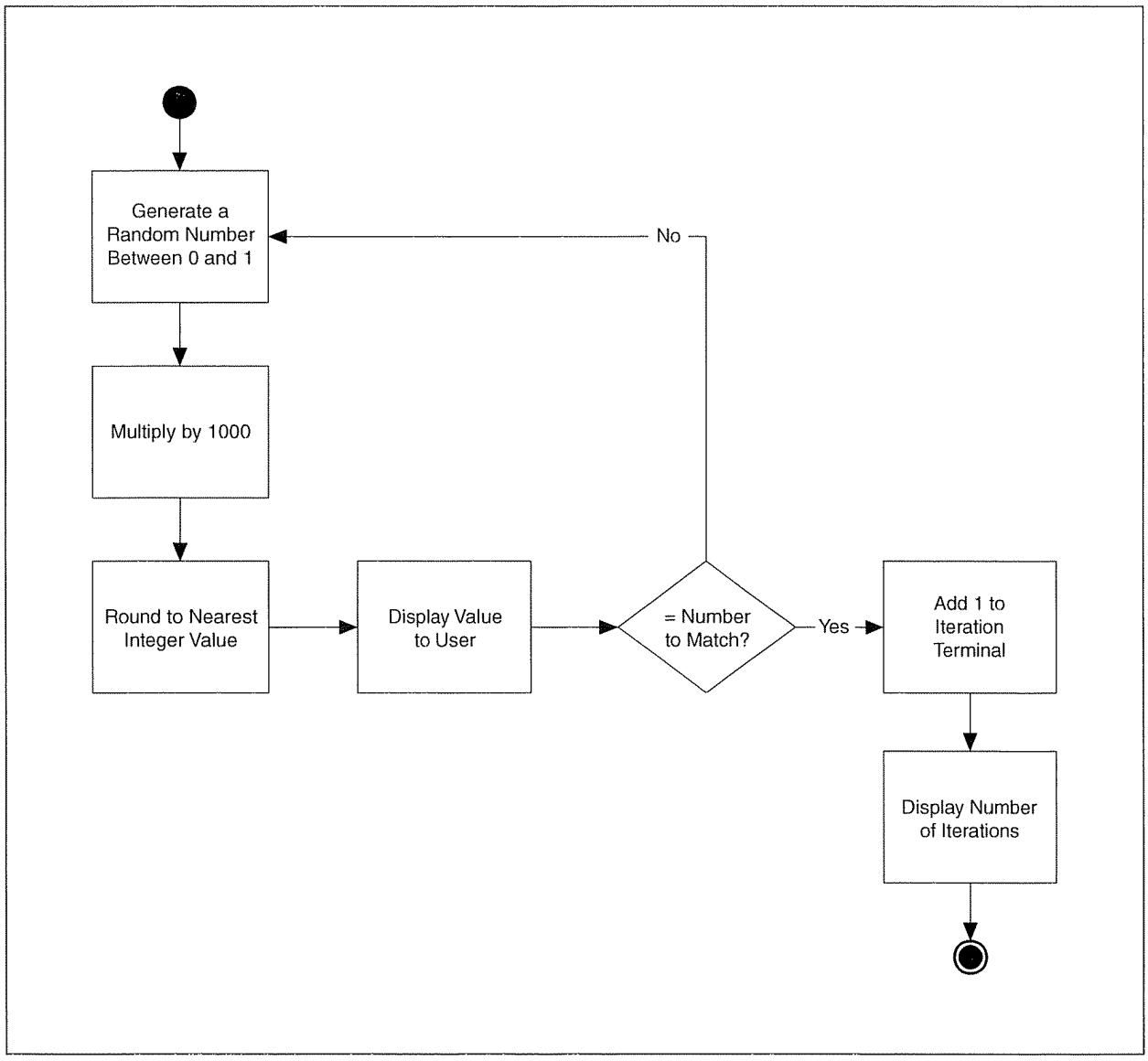


Figura 4-18. Diagrama de flujo de Auto Match

Implementación

Abra un VI nuevo y cree el siguiente panel frontal. Modifique los controles e indicadores como se muestra en el siguiente panel frontal y como se describe en estos pasos.

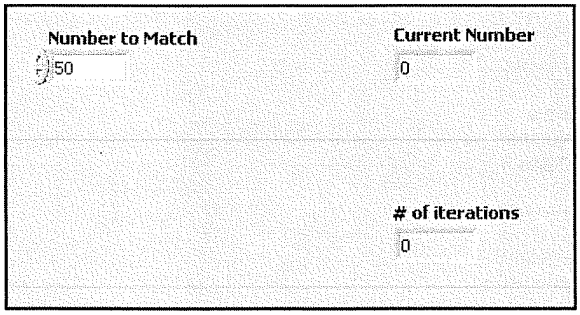


Figura 4-19. Panel frontal del VI Auto Match

1. Abra un VI nuevo.
2. Guarde el VI como `Auto Match.vi` en el directorio `<Exercises>\LabVIEW Basics I\Auto Match`.
3. Cree la entrada **Number to Match**.
   - ❑ Añada un control numérico a la ventana del panel frontal.
   - ❑ Etiquete el control `Number to Match`.
4. Configure el valor predeterminado para el control **Number to Match**.
   - ❑ Configure el control **Number to Match** en `50`.
   - ❑ Haga clic con el botón derecho en el control **Number to Match** y seleccione **Data Operations»Make Current Value Default**.
5. Configure las propiedades del control **Number to Match** para que el intervalo de datos sea de 0 a 1000, el valor de incremento sea 1 y los dígitos de precisión sean 0.
   - ❑ Haga clic con el botón derecho en el control **Number to Match** y seleccione **Data Entry** en el menú contextual. Aparecerá la página **Data Entry** del cuadro de diálogo **Numeric Properties**.
   - ❑ Desmarque la casilla de verificación **Use Default Limits**.
   - ❑ Configure el valor **Minimum** a `0` y seleccione **Coerce** en el menú desplegable **Response to value outside limits**.

- ❑ Configure el valor **Maximum** a `1000` y seleccione **Coerce** en el menú desplegable **Response to value outside limits**.
- ❑ Configure el valor **Increment** a `1` y seleccione **Coerce** en el menú desplegable **Response to value outside limits**.
- ❑ Seleccione la ficha **Display Format**.
- ❑ Seleccione **Floating Point** y cambie el tipo de precisión de **Significant digits** a **Digits of precision**.
- ❑ Escriba `0` en el cuadro de texto **Digits** y haga clic en **OK**.

6. Cree la salida **Current Number**.
   - ❑ Añada un indicador numérico a la ventana del panel frontal.
   - ❑ Etiquete el indicador `Current Number`.
   - ❑ Haga clic con el botón derecho en el indicador y seleccione **Advanced»Synchronous Display**.
7. Configure a 0 los dígitos de precisión para la salida **Number to Match**.
   - ❑ Haga clic con el botón derecho en el indicador **Current Number** y seleccione **Display format** en el menú contextual. Aparecerá la página **Display Format** del cuadro de diálogo **Numeric Properties**.
   - ❑ Seleccione **Floating Point** y cambie el **Precision Type** a **Digits of precision**.
   - ❑ Escriba `0` en el cuadro de texto **Digits** y haga clic en **OK**.
8. Cree la salida **# of iterations**.
   - ❑ Coloque un indicador numérico en el panel frontal.
   - ❑ Etiquete el indicador `# of iterations`.
9. Configure la representación para la salida **# of iterations** en un número entero largo.
   - ❑ Haga clic con el botón derecho en el indicador **# of iterations**.
   - ❑ Seleccione **Representation»I32** en el menú contextual.

Cree el siguiente diagrama de bloques. Siga estos pasos.

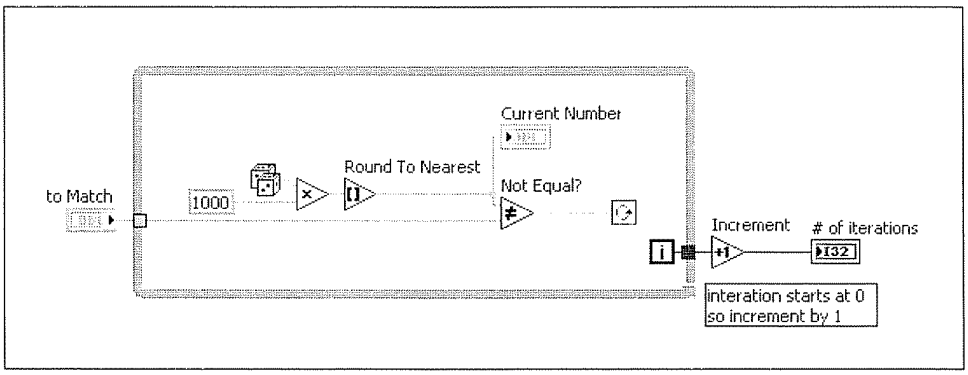


Figura 4-20. Diagrama de bloques del VI Auto Match

10. Genere un número integral aleatorio entre 0 y 1.000.

- ❑ Añada la función Random Number (0-1) al diagrama de bloques. La función Random Number (0-1) produce un número aleatorio entre el 0 y el 1.

- ❑ Añada la función Multiply al diagrama de bloques. La función Multiply multiplica el número aleatorio por la entrada **y** para producir un número aleatorio entre el 0 e **y**.
- ❑ Cablee la salida de la función Random Number a la entrada **x** de la función Multiply.
- ❑ Haga clic con el botón derecho en la entrada **y** de la función Multiply, seleccione **Create»Constant** en el menú contextual, escriba `1000` y pulse la tecla <Intro> para crear una constante numérica.

- ❑ Añada la función Round To Nearest al diagrama de bloques. Esta función redondea el número aleatorio al número entero más cercano.
- ❑ Cablee la salida de la función Multiply a la entrada de la función Round To Nearest. Cablee la salida de la función Round To Nearest al indicador Current Number.

11. Compare el número generado aleatoriamente con el valor del control **Number to Match**.

- ❑ Añada la función Not Equal? al diagrama de bloques. Esta función compara el número aleatorio con Number to Match y devuelve **`True`** si los números no son iguales; de lo contrario, devuelve **`False`**.
- ❑ Cablee la salida de la función Round To Nearest a la entrada **x** de la función Not Equal?.

- ❑ Cablee el control numérico Number to Match con el borde del bucle While. Aparecerá un túnel naranja en el borde del bucle While.
- ❑ Cablee el túnel naranja a la entrada **y** de la función Not Equal?.

12. Repita el algoritmo hasta que la función Not Equal? devuelva True.

- ❑ Añada un bucle While desde la paleta **Structures** al diagrama de bloques.
- ❑ Haga clic con el botón derecho en el terminal condicional y seleccione **Continue if True** en el menú contextual.
- ❑ Cablee la salida de la función Not Equal? al terminal condicional.

13. Muestre al usuario la cantidad de números aleatorios generados añadiendo uno al valor del terminal de iteración.

- ❑ Cablee el terminal de iteración al borde del bucle While. Aparecerá un túnel azul en el borde del bucle While.

Consejo Cada vez que se ejecute el bucle, el terminal de iteración aumentará en uno. Cablee el valor de iteración a la función Increment porque el contaje de iteraciones empieza en `0`. El contaje de iteraciones sale del bucle al completarse.

- ❑ Añada la función Increment al diagrama de bloques. Esta función suma 1 al contaje del bucle While.

14. Guarde el VI.

Prueba

1. Abra el panel frontal.
2. Cambie el número en **Number to Match** a un número que se encuentre en el rango de datos, que es de 0 a 1.000 con un incremento de 1.
3. Ejecute el VI.
4. Cambie **Number to Match** y ejecute el VI de nuevo. **Current Number** se actualiza en cada iteración del bucle porque se encuentra dentro de él. **# of iterations** se actualiza al completarse porque se encuentra fuera del bucle.

5. Para ver cómo el VI actualiza los indicadores, habilite el resaltado de ejecución.

   - ❑ En el diagrama de bloques, haga clic en el botón **Highlight Execution** para habilitar el resaltado de ejecución. El resaltado de la ejecución muestra el movimiento de datos en el diagrama de bloques desde un nodo a otro para que pueda ver cada número según lo genera el VI.

6. Ejecute el VI y observe el flujo de datos.

7. Intente hacer coincidir un número que se encuentre fuera del rango de datos.

8. Cambie **Number to Match** a un número que se encuentre fuera del rango de datos.

   - ❑ Ejecute el VI. LabVIEW convierte el valor fuera de rango al valor más cercano en el rango de datos especificado.

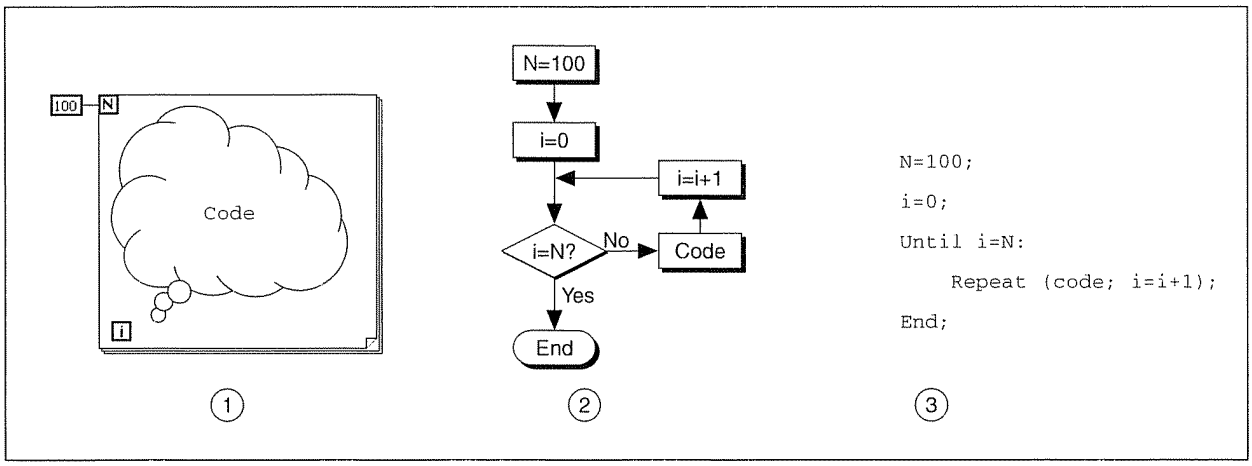
9. Cierre el VI.

Fin del ejercicio 4-2

E. Bucles For

Un bucle For, mostrado abajo, ejecuta un subdiagrama cierto número de veces.

La figura 4-21 muestra un bucle For en LabVIEW, un diagrama de flujo equivalente a la funcionalidad del bucle For, y un ejemplo de pseudocódigo de la funcionalidad del bucle For.



1 Bucle For de LabVIEW 2 Diagrama de flujo 3 Pseudocódigo

Figura 4-21. Bucle For

El bucle For se encuentra en la paleta **Structures**. También puede colocar un bucle While en el diagrama de bloques; haga clic con el botón derecho en el borde del bucle While y seleccione **Replace with For Loop** en el menú contextual para cambiar un bucle While por uno For. El terminal de contaje, que se ve abajo, es un terminal de entrada cuyo valor indica cuántas veces debe repetir el subdiagrama.

El terminal de iteración, que se ve abajo, es un terminal de salida que contiene el numero de iteraciones completadas.

El contaje de iteraciones siempre empieza en cero. Durante la primera iteración, el terminal de iteración devuelve `0`. Si el contaje de iteraciones supera 2.147.483.647 o 2^{31}, el terminal de iteración permanecerá en 2.147.483.647 para todas las iteraciones siguientes. Si necesita contar más de 2.147.483.647 iteraciones, puede utilizar registros de desplazamiento con un rango de número entero mayor.

El bucle For se diferencia del bucle While en que aquél se ejecuta un número determinado de veces. Un bucle While deja de ejecutar el subdiagrama sólo si existe el valor en el terminal condicional.

El siguiente bucle For genera un número aleatorio cada segundo durante 100 segundos y muestra los números aleatorios en un indicador numérico.

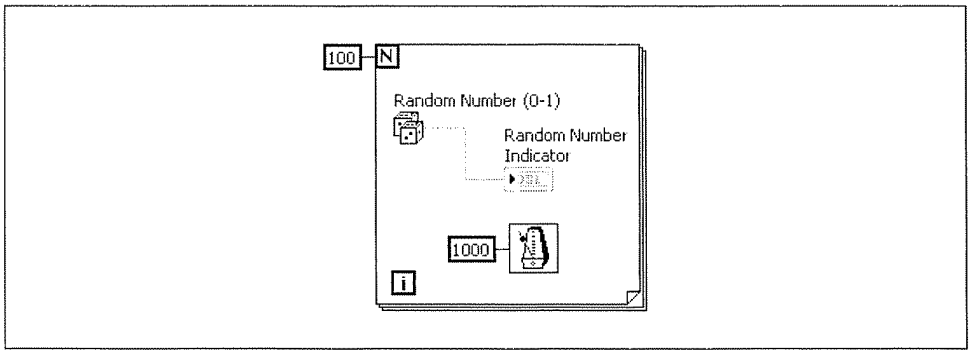


Figura 4-22. Ejemplo de bucle For

Añadir un terminal condicional a un bucle For

Si es necesario, puede añadir un terminal condicional para configurar un bucle For para que se detenga cuando ocurra un error o una condición booleana. Un bucle For con un terminal condicional se ejecuta hasta que ocurra la condición o hasta que se completen todas las iteraciones, lo que suceda primero. Los bucles For que configure para una salida condicional tendrán un símbolo rojo en el terminal de contaje y un terminal condicional en la esquina inferior derecha. Tras configurar el bucle For para salir condicionalmente, el bucle aparece similar a la figura 4-23. El siguiente bucle For genera un número aleatorio cada segundo hasta que transcurran 100 segundos o hasta que el usuario haga clic en el botón de parada.

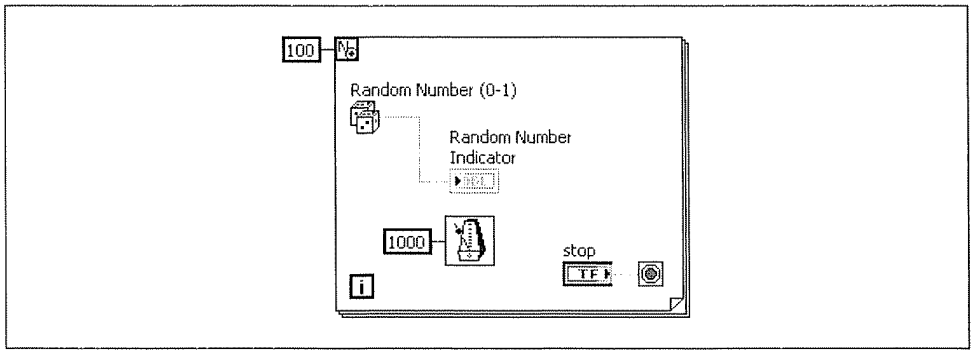


Figura 4-23. Bucle For configurado para una salida condicional

Para añadir un terminal condicional a un bucle For, haga clic con el botón derecho del borde del bucle For y seleccione Conditional Terminal en el menú contextual. A continuación, cablee el terminal condicional y el terminal de contaje.

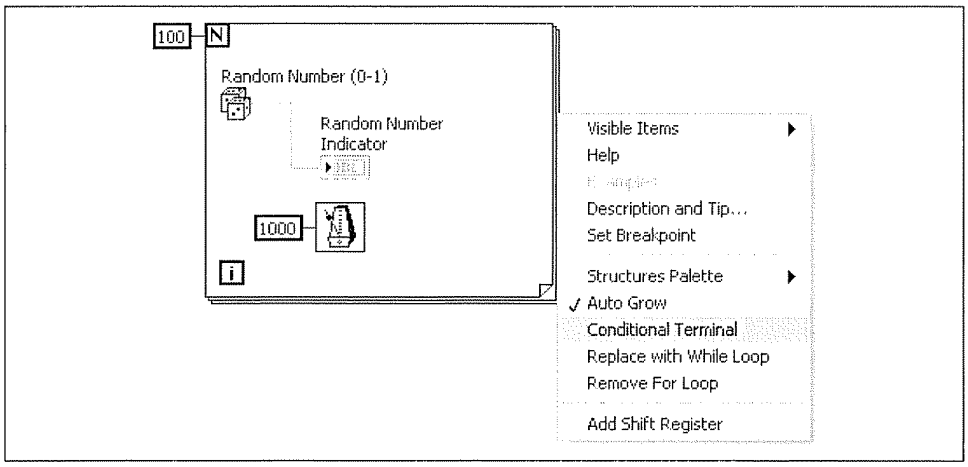
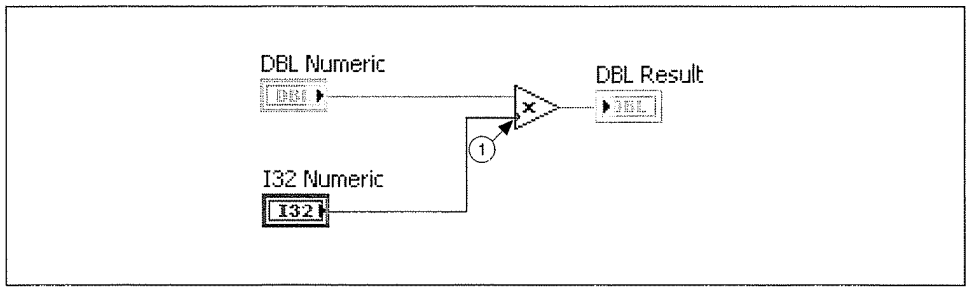


Figura 4-24. Añadir un terminal condicional a un bucle For

Conversión numérica

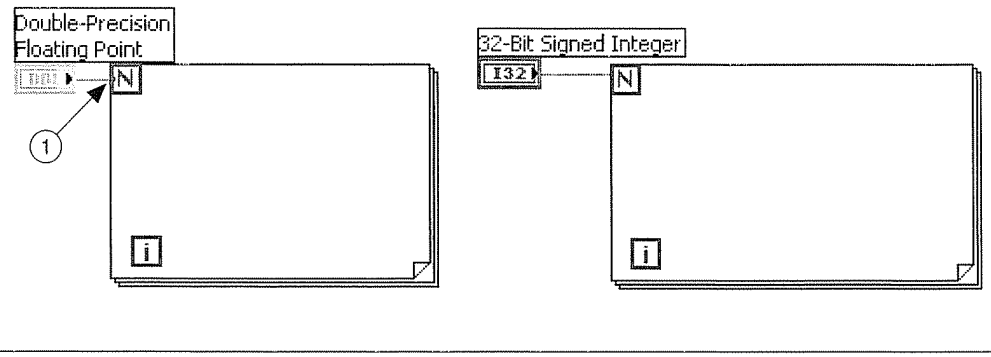
LabVIEW puede representar tipos de datos numéricos como números enteros con signo o sin signo, valores numéricos de coma flotante o valores numéricos complejos, como se describió en la sección *Tipos de datos de LabVIEW* de esta lección. Normalmente, cuando cablea tipos distintos de representaciones en las entradas de una función, ésta devuelve una salida en el formato más grande o amplio. Si usa un número entero con signo con un número entero sin signo, lo convertirá al número entero sin signo. Si usa un número entero sin signo con uno en coma flotante, lo convertirá a coma flotante. Si usa un número entero en coma flotante con un número complejo, lo convertirá al número complejo. Si usa dos números del mismo tipo con distinto ancho de bits, LabVIEW realizará la conversión al mayor de los dos anchos de bits. Si el número de bits es el mismo, LabVIEW elige números enteros sin signo en vez de con signo. Por ejemplo, si cablea un DBL y un I32 a una función Multiply, el resultado es un DBL, como se muestra en la figura 4-25. LabVIEW realizará la conversión al número entero con signo de 32 bits porque utiliza menos bits que el valor numérico de doble precisión y de coma flotante. La entrada inferior de la función Multiply muestra un punto rojo, llamado punto de conversión, que indica que LabVIEW está realizando una conversión de datos.



1 Punto de conversión

Figura 4-25. Ejemplo de conversión numérica

Sin embargo, el terminal de contaje del bucle For funciona del modo opuesto. Si cablea un valor numérico de doble precisión y en coma flotante al terminal de contaje de 32 bits, LabVIEW convierte el valor numérico mayor a un número entero con signo de 32 bits. Aunque la conversión es contraria a los estándares normales de conversión, es necesaria, ya que un bucle For sólo puede ejecutarse un número entero de veces.



1 Punto de conversión

Figura 4-26. Conversión en un bucle For

Ejercicio 4-3 Concepto: bucles While frente a bucles For

Objetivo

Comprender cuándo se usa un bucle While y cuándo uno For.

Descripción

En los siguientes escenarios, decida si usar un bucle While o For.

Escenario 1

Adquiere datos de presión cada segundo durante un minuto.

1. Si usa un bucle While, ¿qué condición necesita para detener el bucle?

2. Si usa un bucle For, ¿cuántas iteraciones necesita ejecutar el bucle?

3. ¿Resulta más fácil implementar un bucle For o un bucle While?

Escenario 2

Adquiere datos de presión hasta que ésta es mayor o igual que 1.400 psi.

1. Si usa un bucle While, ¿qué condición necesita para detener el bucle?

2. Si usa un bucle For, ¿cuántas iteraciones necesita ejecutar el bucle?

3. ¿Resulta más fácil implementar un bucle For o un bucle While?

Escenario 3

Adquiere datos de presión y temperatura hasta que ambos valores son estables durante dos minutos.

1. Si usa un bucle While, ¿qué condición necesita para detener el bucle?

2. Si usa un bucle For, ¿cuántas iteraciones necesita ejecutar el bucle?

3. ¿Resulta más fácil implementar un bucle For o un bucle While?

Escenario 4

Genera una rampa de tensión que comienza en cero, aumentando incrementalmente en 0,5 V cada segundo, hasta que la tensión de salida es igual a 5 V.

1. Si usa un bucle While, ¿qué condición necesita para detener el bucle?

2. Si usa un bucle For, ¿cuántas iteraciones necesita ejecutar el bucle?

3. ¿Resulta más fácil implementar un bucle For o un bucle While?

Respuestas

Escenario 1

Adquiere datos de presión cada segundo durante un minuto.

1. Bucle While: Tiempo = 1 minuto
2. Bucle For: 60 iteraciones
3. Ambos son posibles.

Escenario 2

Adquiere datos de presión hasta que ésta sea 1.400 psi.

1. Bucle While: Presión = 1400 psi
2. Bucle For: no se sabe
3. Un bucle While. Aunque puede añadir un terminal condicional a un bucle For, debe cablear un valor al terminal de contaje. Sin más información, no conoce el valor apropiado para cablear al terminal de contaje.

Escenario 3

Adquiere datos de presión y temperatura hasta que ambos valores son estables durante dos minutos.

1. Bucle While: [(Última Temperatura = Temperatura Previa) durante 2 minutos o más] y [(Última presión = Presión Previa) durante 2 minutos o más]
2. Bucle For: no se sabe
3. Un bucle While. Aunque puede añadir un terminal condicional a un bucle For, debe cablear un valor al terminal de contaje. Sin más información, no conoce el valor apropiado para cablear al terminal de contaje.

Escenario 4

Genera una rampa de tensión que comienza en cero, aumentando incrementalmente en 0,5 V cada segundo, hasta que la tensión de salida es igual a 5 V.

1. Bucle While: Voltaje = 5 V
2. Bucle For: 11 iteraciones
3. Ambos son posibles.

Fin del ejercicio 4-3

F. Temporización de VIs

Cuando un bucle termina de ejecutar una iteración, inmediatamente empieza a ejecutar la siguiente, a menos que llegue a una condición de parada. Normalmente tendrá que controlar la frecuencia o temporización de la iteración. Por ejemplo, si está adquiriendo datos y desea hacerlo cada 10 segundos, necesita poder temporizar las iteraciones del bucle para que ocurran cada 10 segundos.

Aunque no necesite que la ejecución ocurra con cierta frecuencia, debe facilitar al procesador tiempo para completar otras tareas, como responder a la interfaz del usuario. Esta sección explica métodos para temporizar los bucles.

Funciones de espera

Coloque una función de espera dentro de un bucle para que un VI se desactive durante cierto tiempo. Esto permite al procesador abordar otras tareas durante el tiempo de espera. Las funciones de espera usan el reloj de milisegundos del sistema operativo.

La función Wait Until Next ms Multiple controla un contador de milisegundos y espera hasta que éste alcance un múltiplo de la cantidad que especifique. Use esta función para sincronizar actividades. Sitúe esta función en un bucle para controlar la velocidad de ejecución de éste. Para que esta función sea efectiva, el tiempo de ejecución del código debe ser inferior al tiempo especificado para esta función. El tiempo de ejecución de la primera iteración del bucle es indeterminado.

La función Wait (ms) espera hasta que el contador de milisegundos cuente una cantidad igual a la entrada que especifique. Esta función garantiza que la velocidad de ejecución del bucle se corresponda al menos con el valor que especifique a la entrada de la función.

Nota El VI Express Time Delay actúa de modo similar a la función Wait (ms) pero con clusters de error integrados. Consulte la lección 3, *Resolución de problemas y depuración de VIs*, para obtener información adicional acerca de los clusters de error.

Tiempo transcurrido

En algunos casos, resulta útil determinar cuánto tiempo transcurre después de un punto en su VI. El VI Express Elapsed Time indica el tiempo transcurrido tras el instante de inicio especificado. Este VI Express controla el tiempo mientras que el VI sigue ejecutándose. Este VI Express no facilita al procesador el tiempo para completar otras tareas. Utilizará el VI Express Elapsed Time en el proyecto del curso Weather Station.

G. Transferencia de datos iterativa

Al programar con bucles, a menudo debe acceder a datos de iteraciones anteriores del bucle en LabVIEW. Por ejemplo, si está adquiriendo un dato en cada iteración de un bucle y debe hallar el promedio cada cinco datos, debe conservar los datos de iteraciones anteriores del bucle. Los registros de desplazamiento transfieren valores de una iteración del bucle a la siguiente.

Nota Los nodos de realimentación son otro método para conservar información de una iteración anterior. Consulte el tema *Feedback Node* de la *Ayuda de LabVIEW* para obtener información adicional acerca de nodos de realimentación.

Los registros de desplazamiento son similares a las variables estáticas en lenguajes de programación basados en texto.

Use registros de desplazamiento cuando desee pasar valores de iteraciones anteriores a la siguiente iteración a través del bucle. Un registro de desplazamiento aparece como un par de terminales, mostrados abajo, directamente opuestos entre sí en los lados verticales del borde del bucle.

El terminal de la derecha del bucle contiene una flecha hacia arriba y almacena datos al completar una iteración. LabVIEW transfiere los datos conectados al lado derecho del registro en la siguiente iteración. Después de que se ejecute el bucle, el terminal del lado derecho del bucle devuelve el último valor almacenado en el registro de desplazamiento.

Cree un registro de desplazamiento haciendo clic con el botón derecho en el borde izquierdo o derecho de un bucle y seleccionando **Add Shift Register** en el menú contextual.

Un registro de desplazamiento transfiere cualquier tipo de datos y cambia automáticamente al tipo de datos del primer objeto cableado al registro de desplazamiento. Los datos que cablee a los terminales de cada registro de desplazamiento deben ser del mismo tipo.

Puede añadir más de un registro de desplazamiento a un bucle. Si tiene varias operaciones que usan valores de iteraciones previas dentro del bucle, use varios registros de desplazamiento para almacenar los valores de datos de los distintos procesos en la estructura, como se muestra en la siguiente figura.

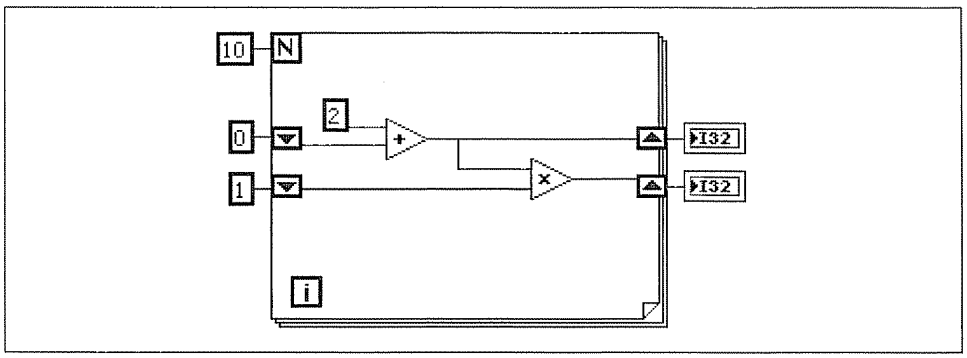
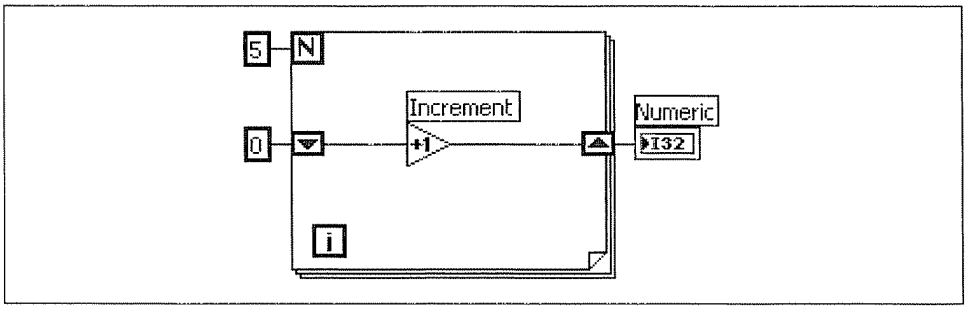


Figura 4-27. Uso de varios registros de desplazamiento

Inicialización de registros de desplazamiento

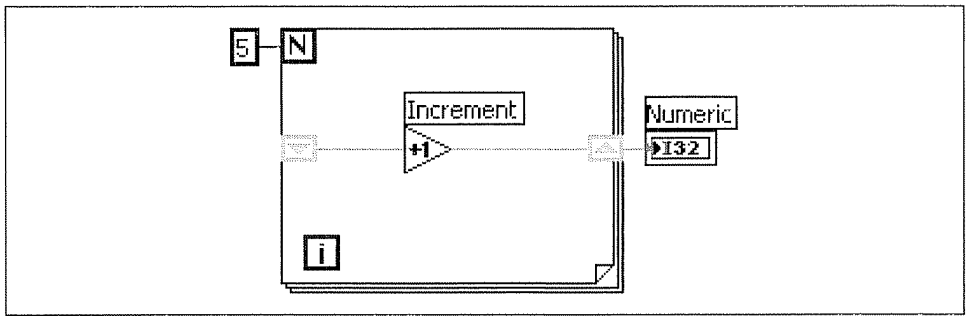
Al inicializar un registro de desplazamiento, se restablece el valor que transmite el registro de desplazamiento a la primera iteración del bucle cuando se ejecuta un VI. Inicialice un registro de desplazamiento cableando un control o constante al terminal del registro de desplazamiento en el lado izquierdo del bucle, como se muestra en el siguiente diagrama de bloques.



En el diagrama de bloques anterior, el bucle For se ejecuta cinco veces, incrementando el valor que lleva el registro de desplazamiento en uno cada vez. Tras cinco iteraciones del bucle For, el registro de desplazamiento pasa el valor final, `5`, al indicador y el VI finaliza su ejecución. Cada vez que ejecute el VI, el registro de desplazamiento empieza con un valor de `0`.

Si no inicializa el registro de desplazamiento, el bucle usa el valor escrito en el registro de desplazamiento cuando se ejecutó por última vez el bucle o, si éste nunca se ejecutó, usa el valor predeterminado del tipo de dato.

Use un registro de desplazamiento sin inicializar para conservar la información de estado entre las siguientes ejecuciones de un VI. El siguiente diagrama de bloques muestra un registro de desplazamiento sin inicializar.

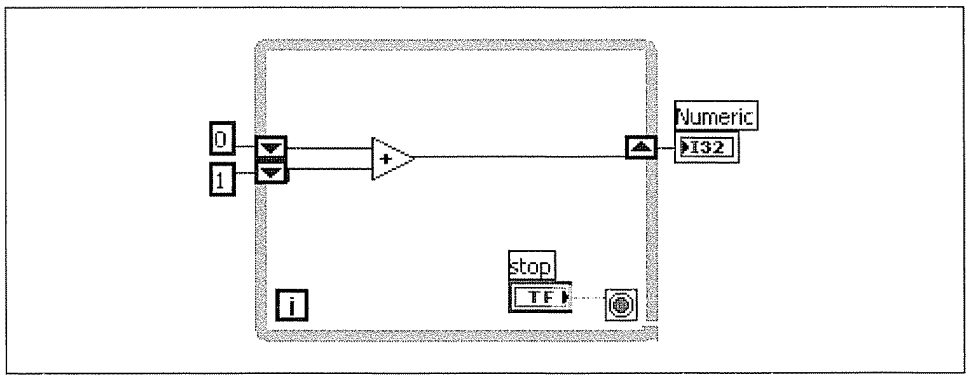


En el diagrama de bloques anterior, el bucle For se ejecuta cinco veces, incrementando el valor que lleva el registro de desplazamiento en uno cada vez. La primera vez que ejecute el VI, el registro de desplazamiento empieza con un valor de `0`, que es el valor predeterminado de un número entero de 32 bits. Tras cinco iteraciones del bucle For, el registro de desplazamiento pasa el valor final, `5`, al indicador y el VI finaliza su ejecución. La siguiente vez que ejecute el VI, el registro de desplazamiento comenzará con un valor de `5`, que era el último valor de la ejecución anterior. Tras cinco iteraciones del bucle For, el registro de desplazamiento pasa el valor final, `10`, al indicador. Si ejecuta de nuevo el VI, el registro de desplazamiento empieza con un valor de `10`, y así sucesivamente. Los registros de desplazamiento sin inicializar conservan el valor de la iteración anterior hasta que cierre el VI.

Registros de desplazamiento apilados

Los registros de desplazamiento apilados permiten acceder a datos de iteraciones anteriores al bucle. Los registros de desplazamiento apilados almacenan valores de varias iteraciones anteriores y llevan esos valores a las siguientes iteraciones. Para crear un registro de desplazamiento apilado, haga clic con el botón derecho en el terminal izquierdo y seleccione **Add Element** en el menú contextual.

Los registros de desplazamiento apilados sólo pueden utilizarse en el lado izquierdo del bucle, porque el terminal derecho transfiere los datos generados sólo desde la iteración actual a la siguiente, como se muestra en este diagrama de bloques.



Si añade otro elemento al terminal izquierdo del diagrama de bloques anterior, los valores de las dos últimas iteraciones se transmiten a la siguiente iteración, almacenándose el valor de iteración más reciente en el registro de desplazamiento superior. El terminal inferior almacena los datos que le llegan de la iteración anterior.

Ejercicio 4-4 VI Average Temperature

Objetivo

Usar un bucle While y registros de desplazamiento para hallar el promedio de datos.

Escenario

El VI Temperature Monitor adquiere y muestra la temperatura. Modifique el VI para hallar el promedio de las tres últimas mediciones de temperatura y mostrar el promedio móvil en el gráfico tipo "waveform chart".

Diseño

La figura 4-28 y la figura 4-29 muestran el panel frontal y el diagrama de bloques del VI Temperature Monitor.

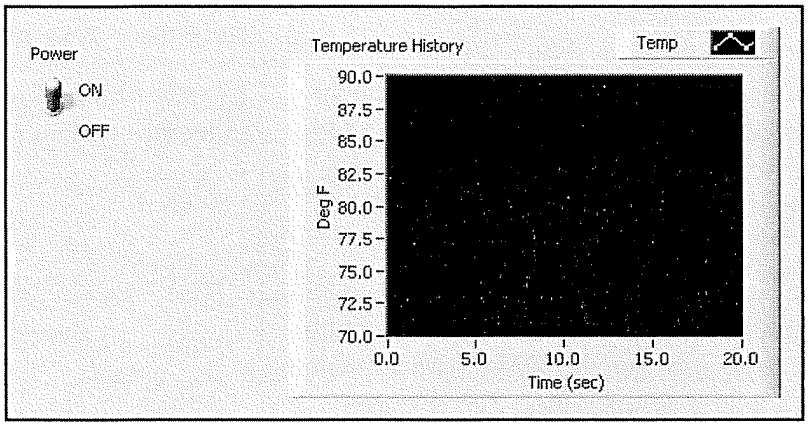


Figura 4-28. Panel frontal del VI Temperature Monitor

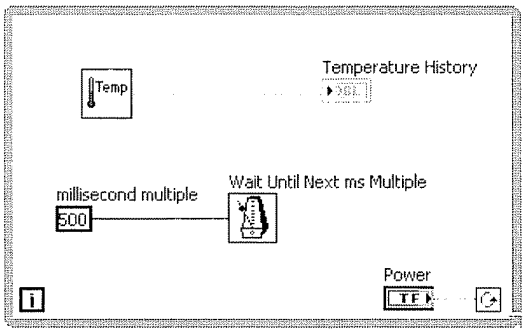


Figura 4-29. Diagrama de bloques del VI Temperature Monitor

Para modificar este VI, debe conservar los valores de temperatura de las dos iteraciones anteriores y hallar el promedio de los valores. Use un registro de desplazamiento con un elemento adicional para conservar los datos de las dos iteraciones anteriores. Inicialice el registro de desplazamiento con una lectura desde el sensor de temperatura. Registre gráficamente sólo la temperatura media.

Implementación

1. Pruebe el VI. Si dispone del hardware, siga las instrucciones de la columna **Con hardware instalado**. De lo contrario, siga las instrucciones de la columna **Sin hardware instalado**.

Con hardware instalado	Sin hardware instalado
Abra el VI Temperature Monitor en el directorio `<Exercises>\LabVIEW Basics I\Average Temperature`.	Abra el VI Temperature Monitor (Demo) en el directorio `<Exercises>\LabVIEW Basics I\No Hardware Required\Average Temperature`.
Seleccione **File»Save As** y renombre el VI como `Average Temperature.vi` en el directorio `<Exercises>\LabVIEW Basics I\Average Temperature`.	Seleccione **File»Save As** y renombre el VI como `Average Temperature.vi` en el directorio `<Exercises>\LabVIEW Basics I\No Hardware Required\Average Temperature`.
En el Accesorio de señales DAQ, cambie el interruptor de ruido del sensor de temperatura a la posición On. Este interruptor introduce ruido en la lectura de la temperatura.	Ejecute el VI. Observe la variación en la lectura de la temperatura simulada.
Ejecute el VI.	
Coloque el dedo en el sensor de temperatura del Accesorio de señales DAQ, para aumentar la lectura de la temperatura. Puede mover rápidamente el dedo por el sensor para aumentar la lectura aún más con la fricción. Observe el elevado número de picos en la lectura.	

2. Detenga el VI cambiando el estado del interruptor **Power** en el panel frontal. Observe que el interruptor **Power** cambia inmediatamente al estado **On**. La acción mecánica del interruptor controla este comportamiento.

En los siguientes pasos, modifique el VI para reducir el número de picos de temperatura.

3. Abra el diagrama de bloques.

4. Modifique el diagrama de bloques como en la figura 4-30.

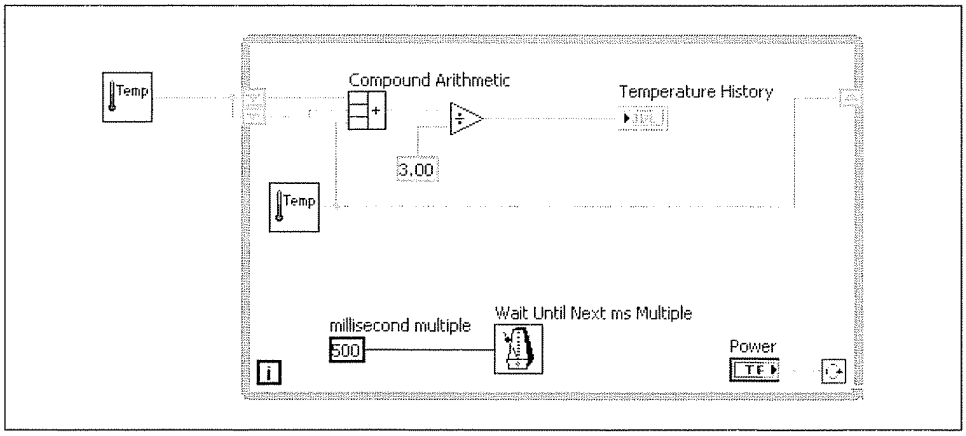


Figura 4-30. Diagrama de bloques del VI Average Temperature

- ❑ Haga clic con el botón derecho en el borde derecho o izquierdo del bucle While y seleccione **Add Shift Register** en el menú contextual para crear un registro de desplazamiento.
- ❑ Haga clic con el botón derecho en el terminal izquierdo del registro de desplazamiento y seleccione **Add Element** en el menú contextual para añadir un elemento al registro de desplazamiento.

- ❑ Pulse la tecla <Ctrl> mientras hace clic en el VI Thermometer y arrástrelo fuera del bucle While para crear una copia del subVI.

  El VI Thermometer devuelve una medida de temperatura desde el sensor de temperatura e inicializa los registros de desplazamiento izquierdos antes de que se inicie el bucle.

- ❑ Coloque la función Compound Arithmetic en el diagrama de bloques.
  - Configure esta función para devolver la suma de la temperatura actual y las dos lecturas anteriores de la temperatura.
  - Use la herramienta de posicionamiento para cambiar el tamaño de la función Compound Arithmetic para tener tres terminales a la izquierda.

- ❑ Coloque la función Divide en el diagrama de bloques. Esta función devuelve el promedio de las tres últimas lecturas de temperatura.
- ❑ Cablee las funciones como en la figura 4-30.
- ❑ Haga clic con el botón derecho en la entrada **y** de la función Divide y seleccione **Create»Constant**.
- ❑ Introduzca 3 y pulse la tecla <Intro>.

5. Guarde el VI.

Prueba

1. Ejecute el VI.
2. Si tiene el hardware instalado, coloque el dedo en el sensor de temperatura del Accesorio de Señales DAQ, para aumentar la lectura de la temperatura.

   Durante cada iteración en el bucle While, el VI Thermometer toma una medida de temperatura. El VI añade este valor a las dos últimas mediciones almacenadas en los terminales izquierdos del registro de desplazamiento. El VI divide el resultado entre tres para averiguar la media de las tres mediciones (la medida actual más las dos anteriores). El VI muestra el promedio en el gráfico tipo "waveform chart". Observe que el VI inicializa el registro de desplazamiento con una medida de temperatura.
3. Detenga el VI cambiando el estado del interruptor **Power** en el panel frontal.
4. Cierre el VI.

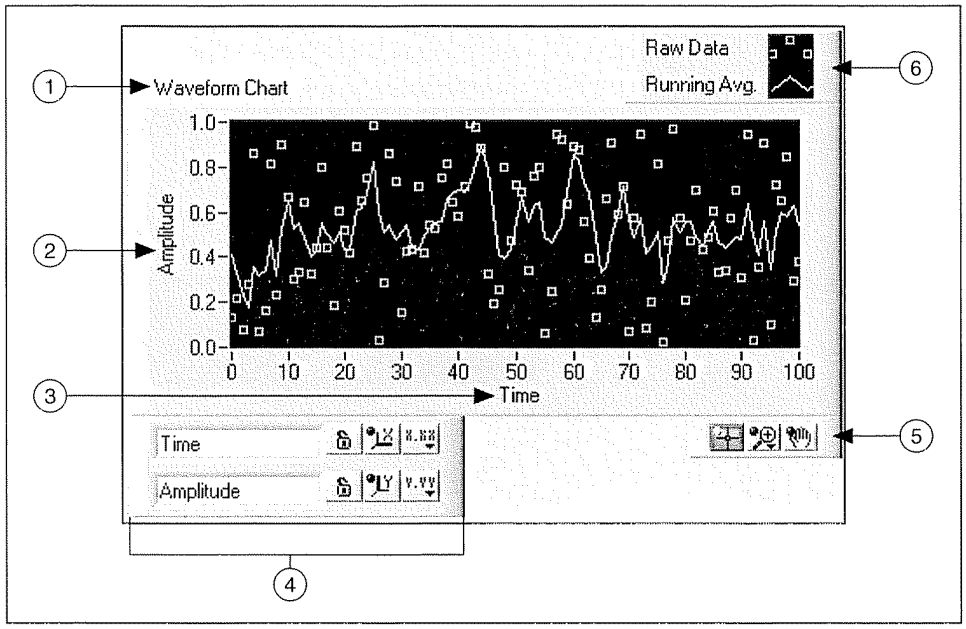
Fin del ejercicio 4-4

H. Representación de datos

Ya utilizó gráficos y tablas para representar datos sencillos. Esta sección explica más en profundidad cómo usar y personalizar tablas y gráficos.

Gráficos tipo "waveform chart"

El gráfico tipo "waveform chart" es un tipo especial de indicador numérico que muestra una o más curvas de datos, normalmente adquiridos a una velocidad constante. Los gráficos tipo "waveform chart" pueden mostrar una o varias curvas. La figura 4-31 muestra los elementos de un gráfico tipo "waveform chart" de varias curvas. Aparecen dos curvas: Raw Data (datos en bruto) y Running Avg (promedio móvil).



1	Etiqueta	3	Escala X	5	Paleta del gráfico
2	Escala Y	4	Leyenda de la escala	6	Leyenda de la curva

Figura 4-31. Gráficos tipo "waveform chart"

Puede configurar cómo se actualiza el gráfico para mostrar nuevos datos. Haga clic con el botón derecho en el gráfico y seleccione **Advanced» Update Mode** desde el menú contextual para configurar el modo de actualización del gráfico. El gráfico utiliza los siguientes modos para mostrar datos:

- **Strip Chart**: muestra los datos móviles desplazándose continuamente de izquierda a derecha por el gráfico con los datos antiguos a la izquierda y con los nuevos a la derecha. Un strip chart es similar a un registrador de cinta de papel. **Strip Chart** es el modo de actualización predeterminado.

- **Scope Chart**: muestra un elemento de datos, como un impulso o una onda, desplazándose parcialmente por el gráfico de izquierda a derecha. Para cada nuevo valor, el gráfico traza el valor a la derecha del último valor. Cuando la curva llega al borde derecho de la zona de trazado, LabVIEW borra la curva y empieza a trazar de nuevo desde el borde izquierdo. La visualización de un scope chart es similar a la de un osciloscopio.
- **Sweep Chart**: funciona de modo parecido a un scope chart salvo que muestra los datos antiguos a la derecha y los nuevos a la izquierda separados por una línea vertical. LabVIEW no borra la curva en un sweep chart cuando ésta llega al borde derecho de la zona de trazado. Un sweep chart es similar a la visualización de un electrocardiograma.

La figura 4-32 muestra un ejemplo de cada modo de actualización del gráfico. El scope chart y el sweep chart tienen visualizaciones de retrazado parecidas a un osciloscopio. Como retrazar una curva requiere menos esfuerzo, el scope chart y el sweep chart muestran curvas mucho más rápidamente que el strip chart.

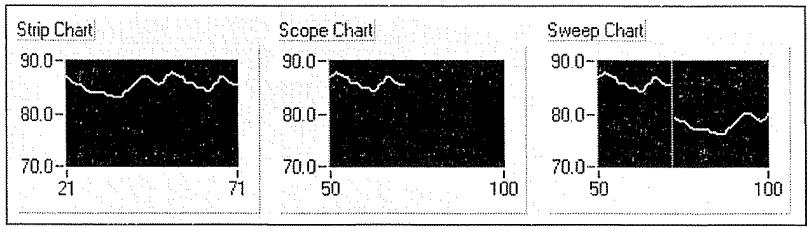


Figura 4-32. Modos de actualización de gráficos

Cableado de gráficos

Puede cablear una salida escalar directamente a un gráfico tipo "waveform chart". El terminal del gráfico tipo "waveform chart" de la figura 4-33 coincide con el tipo de datos de entrada.

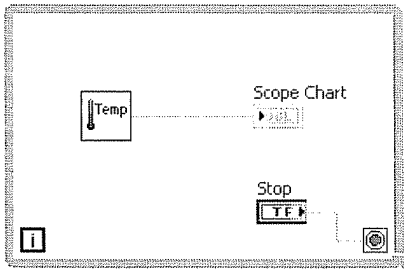


Figura 4-33. Cableado de una sola curva a un gráfico tipo "waveform chart"

Los gráficos tipo "waveform chart" pueden mostrar varias curvas juntas usando la función Bundle situada en la paleta **Cluster, Class & Variant**. En la figura 4-34, la función Bundle une las salidas de los tres VIs en la curva del gráfico tipo "waveform chart".

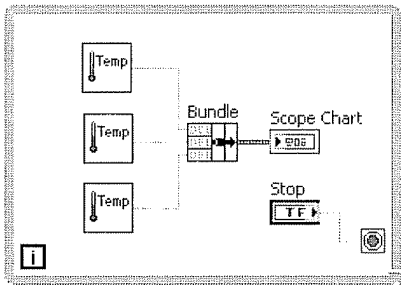
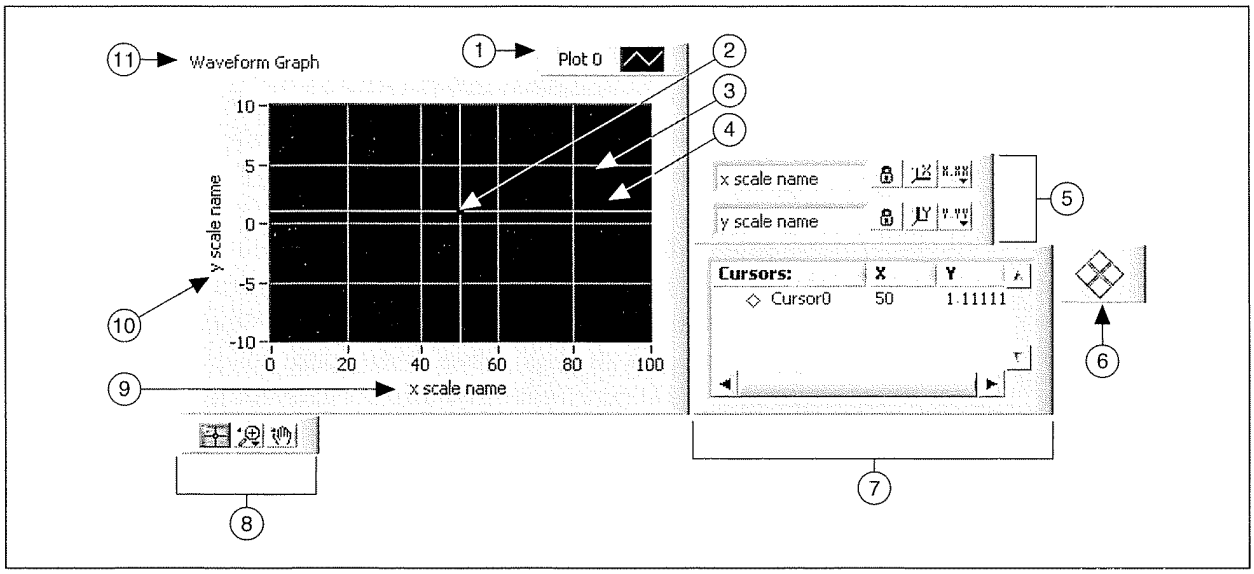


Figura 4-34. Cableado de varias curvas a un gráfico tipo "waveform chart"

El terminal del gráfico tipo "waveform chart" cambia para coincidir con la salida de la función Bundle. Para añadir más curvas, use la herramienta de posicionamiento para cambiar el tamaño de la función Bundle. Consulte la lección 5, *Relacionar datos*, para obtener información adicional acerca de la función Bundle.

Gráficos tipo "Waveform graph"

Los VIs con un gráfico normalmente acumulan los datos en un array y después los trazan en el gráfico. La figura 4-35 muestra los elementos de un gráfico tipo "waveform graph".



1	Leyenda de la curva	5	Leyenda de la escala	9	Escala X
2	Cursor	6	Desplazador del cursor	10	Escala Y
3	Marca de cuadrícula	7	Leyenda del cursor	11	Etiqueta
4	Marca de minicuadrícula	8	Paleta del gráfico		

Figura 4-35. Gráfico tipo "Waveform Graph"

Los gráficos situados en la paleta Graph Indicators incluyen los tipos "waveform graph" y "XY graph". El tipo "waveform graph" sólo representa funciones de una única variable, como en $y = f(x)$, con puntos distribuidos uniformemente en el eje x, como las formas de onda adquiridas variables con el tiempo.

Los gráficos XY muestran cualquier conjunto de puntos, muestreados uniformemente o no.

Cambie el tamaño de la leyenda de la curva para ver varias curvas. Use varias curvas para ahorrar espacio en el panel frontal y comparar curvas. Los gráficos tipo "XY" y "waveform" se adaptan automáticamente a varias curvas.

Gráficos tipo "Waveform graph" de una sola curva

El gráfico tipo "waveform graph" acepta varios tipos de datos para gráficos de una sola curva. El gráfico tipo "graph" acepta un solo array de valores, interpreta los datos como puntos del gráfico e incrementa el índice *x* en uno empezando por $x = 0$. Asimismo acepta un cluster con un valor inicial *x*, un valor delta *x* y un array de datos *y*. Por último, también acepta el tipo de datos de forma de onda, que contiene los datos, tiempo de inicio y el valor delta *t* de una forma de onda.

Consulte el VI Waveform Graph de `labview\examples\general\graphs\gengraph.llb` para ver ejemplos de los tipos de datos que acepta un gráfico tipo "waveform graph".

Gráficos tipo "Waveform graph" de varias curvas

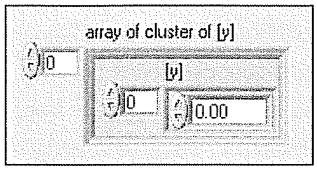
El gráfico tipo "waveform graph" acepta varios tipos de datos para mostrar varias curvas. Acepta un array 2D de valores, donde cada fila de ésta es una sola curva. El gráfico interpreta los datos como puntos del mismo e incrementa el índice *x* en uno, empezando por $x = 0$. Cablee un tipo de datos de array 2D al gráfico tipo "graph", haga clic en éste con el botón derecho y seleccione **Transpose Array** en el menú contextual para tratar cada columna del array como una curva. Esto resulta especialmente útil cuando muestrea varios canales de un dispositivo DAQ, porque éste puede devolver los datos como arrays 2D con cada canal almacenado como una columna distinta.

Consulte el gráfico (Y) Multi Plot 1 en el VI Waveform Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos.

El gráfico tipo "waveform graph" también acepta un cluster con un valor inicial *x*, un valor delta *x* y un array 2D de datos *y*. El gráfico interpreta los datos *y* como puntos del mismo e incrementa el índice *x* por el valor delta *x*,

empezando por el valor *x* inicial. Este tipo de datos resulta útil para mostrar varias señales que se muestrean a la misma velocidad regular. Consulte el gráfico (Xo = 10, dX = 2, Y) Multi Plot 2 en el VI Waveform Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos.

El gráfico tipo "waveform graph" acepta un array de curvas donde el array contiene clusters. Cada cluster contiene un array 1D que alberga los datos *y*. El array interno describe los puntos de una curva y el externo tiene un cluster para cada curva. El siguiente panel frontal muestra este array de clusters **y**.



Use un array de curva en lugar de un array 2D si el número de elementos de cada curva es distinto. Por ejemplo, cuando muestrea datos de varios canales con distintos tiempos para cada uno, use esta estructura de datos en lugar de un array 2D porque cada fila de un array 2D debe tener el mismo número de elementos. El número de elementos de los arrays interiores de un array de clusters puede variar. Consulte el gráfico (Y) Multi Plot 2 en el VI Waveform Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos.

El gráfico tipo "waveform graph" acepta un cluster de un valor inicial *x*, un valor delta *x* y un array que contiene clusters. Cada cluster contiene un array 1D que alberga los datos *y*. Use la función Bundle para unir los arrays en clusters y use la función Build Array para crear los clusters resultantes en un array. También puede utilizar la función Build Cluster Array, que crea arrays de clusters que contienen las entradas que especifique. Consulte el gráfico (Xo = 10, dX = 2, Y) Multi Plot 3 en el VI Waveform Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos.

El gráfico tipo "waveform graph" acepta un array de clusters de un valor *x*, un valor delta *x* y un array de datos *y*. Éste es el más general de los tipos de datos de gráficos tipo "waveform graph" de varias curvas, porque puede indicar un único punto de inicio e incrementar la escala x de cada curva. Consulte el gráfico (Xo = 10, dX = 2, Y) Multi Plot 1 en el VI Waveform Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos.

El gráfico tipo "waveform graph" también acepta el tipo de datos dinámicos, que se usa con los VIs Express. Además de los datos asociados con una señal, el tipo de datos dinámicos incluye atributos con información sobre la señal, como el nombre de la señal o la fecha y la hora en que se adquirieron los datos. Los atributos especifican cómo aparece la señal en el gráfico. Cuando el tipo de datos dinámico incluye varios canales, el gráfico muestra una curva para cada canal y da formato automáticamente a la leyenda de la curva y la marca de hora de la escala x.

Gráficos XY de una sola curva

El gráfico XY acepta tres tipos de datos para gráficos XY de una sola curva. El gráfico XY acepta un cluster que contiene un array *x* y un array *y*. Consulte el gráfico (X and Y arrays) Single Plot en el VI XY Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos.

El gráfico XY también acepta un array de puntos, donde un punto es un cluster que contiene un valor *x* y un valor *y*. Consulte el gráfico (Array of Pts) Single Plot en el VI XY Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos. El gráfico XY también acepta un array de datos complejos, donde la parte real se traza en el eje x y la parte imaginaria en el eje y.

Gráficos XY de varias curvas

El gráfico XY acepta tres tipos de datos para mostrar varias curvas. Acepta un array de curvas, donde una curva es un cluster que contiene un array *x* y un array *y*. Consulte el gráfico (X and Y arrays) Multi Plot en el VI XY Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos.

El gráfico XY también acepta un array de clusters de curvas, siendo una curva un array de puntos. Un punto es un cluster que contiene un valor *x* y un valor *y*. Consulte el gráfico (Array of Pts) Multi Plot en el VI XY Graph de `labview\examples\general\graphs\gengraph.llb` para ver un ejemplo de un gráfico que acepta este tipo de datos. El gráfico XY también acepta un array clusters de curvas, siendo una curva un array de datos complejos, donde la parte real se traza en el eje x y la parte imaginaria en el eje y.

Ejercicio 4-5 VI Temperature Multiplot

Objetivo

Representar varios conjuntos de datos en un solo gráfico tipo "waveform chart" y personalizar su vista.

Escenario

Modifique el VI del ejercicio 4-4 para representar la temperatura actual y el promedio móvil en el mismo gráfico. Además, el usuario debe poder examinar una parte de la curva mientras los datos se están adquiriendo.

Diseño

La figura 4-36 muestra el panel frontal del VI (VI Average Temperature) existente y la figura 4-37 muestra el diagrama de bloques.

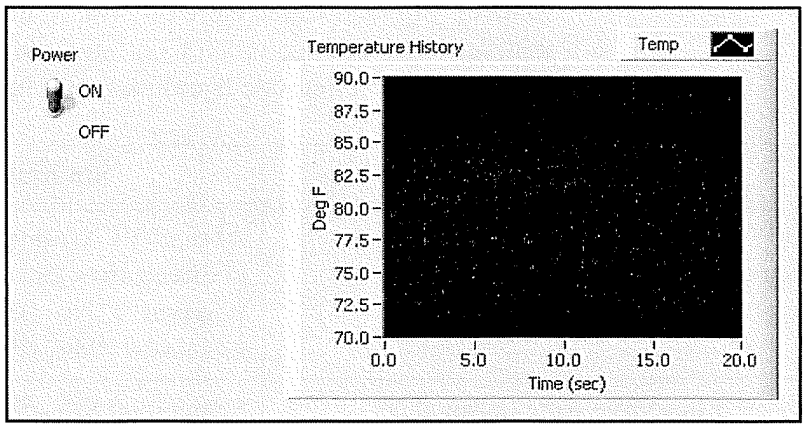


Figura 4-36. Panel frontal del VI Average Temperature

Para que el usuario pueda examinar una parte de la curva mientras los datos se están adquiriendo, muestre la leyenda escalas y la paleta del gráfico tipo "waveform chart". Asimismo, expanda la leyenda para mostrar curvas adicionales.

Para modificar el diagrama de bloques de la figura 4-37, debe modificar el terminal del gráfico para aceptar varios datos. Use una función Bundle para combinar la temperatura media y la actual en un cluster y cablearlo al terminal del gráfico **Temperature History**.

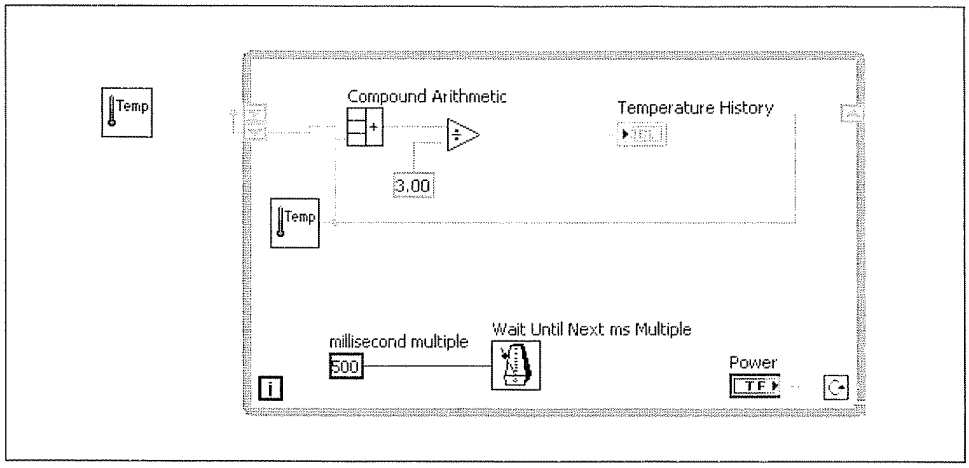


Figura 4-37. Diagrama de bloques del VI Average Temperature

Implementación

1. Abra el VI creado en el ejercicio 4-4. Si dispone del hardware, siga las instrucciones de la columna **Con hardware instalado**. De lo contrario, siga las instrucciones de la columna **Sin hardware instalado**.

Con hardware instalado	Sin hardware instalado
Abra el VI Average Temperature en el directorio `<Exercises>\LabVIEW Basics I\Average Temperature`.	Abra el VI Average Temperature en el directorio `<Exercises>\LabVIEW Basics I\No Hardware Required\Average Temperature`.
Seleccione **File»Save As** y llame al VI `Temperature Multiplot.vi` en el directorio `<Exercises>\LabVIEW Basics I\Temperature Multiplot`.	Seleccione **File»Save As** y llame al VI `Temperature Multiplot.vi` en el directorio `<Exercises>\LabVIEW Basics I\No Hardware Required\Temperature Multiplot`.

Consejo Seleccione la opción **Substitute Copy for Original** para cerrar el VI Average Temperature y trabajar en el VI Temperature Multiplot. Puede crear el directorio si no existe.

En las instrucciones inferiores, modificará el diagrama de bloques similar para incluir la funcionalidad mostrada en la figura 4-38. Modifique primero el diagrama de bloques y luego el panel frontal.

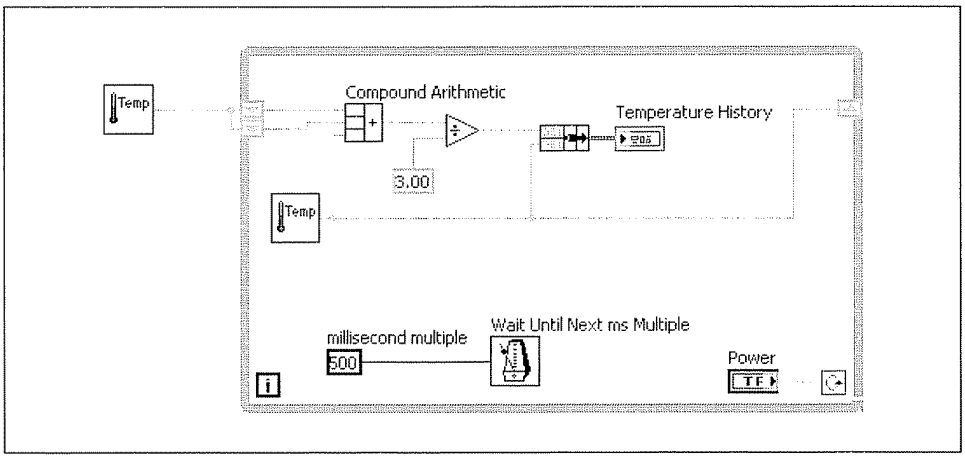


Figura 4-38. Diagrama de bloques del VI Temperature Multiplot

2. Abra el diagrama de bloques.

3. Conecte las temperaturas actual y la media al terminal de gráfico **Temperature History**.

   - ❑ Elimine el cable que conecta la función Divide al terminal del gráfico **Temperature History**.

   - ❑ Añada una función Bundle entre la función Divide y el indicador del gráfico **Temperature History**. Si es necesario, amplíe el bucle While para hacer espacio.

   - ❑ Cablee la salida de la función Divide a la entrada superior de la función Bundle.

   - ❑ Cablee la temperatura actual a la entrada inferior de la función Bundle. La temperatura actual es la salida del subVI Thermometer dentro del bucle While.

   - ❑ Cablee la salida de la función Bundle del indicador del gráfico **Temperature History**.

En los siguientes pasos modifique el panel frontal para obtener una apariencia similar al de la figura 4-39.

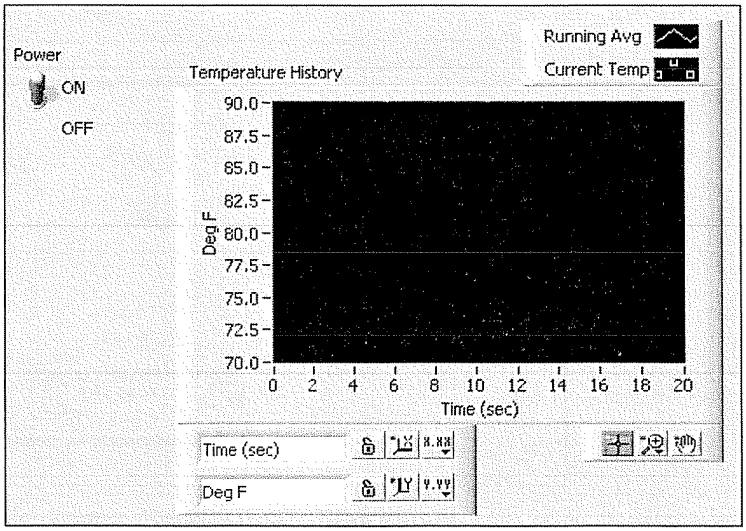


Figura 4-39. Panel frontal del VI Temperature Multiplot

4. Abra el panel frontal.

5. Muestre ambas curvas en la leyenda de curvas del gráfico tipo "waveform chart".

   - ❑ Use la herramienta de posicionamiento para cambiar el tamaño de la leyenda de curvas para mostrar dos objetos, mediante el nodo de cambio de tamaño de la parte media superior.
   - ❑ Llame a la curva superior `Running Avg`.
   - ❑ Llame a la curva inferior `Current Temp`.
   - ❑ Cambie el tipo de curva de Current Temp. Use la herramienta de posicionamiento para seleccionar la curva en la leyenda de curvas y elija la representación que desee.

Consejo El orden de las curvas que aparecen en la leyenda de curvas es el mismo que el de los elementos cableados a la función Bundle del diagrama de bloques.

6. Muestre la leyenda de escalas y la paleta de gráfico del gráfico tipo "waveform chart".

   - ❑ Haga clic con el botón derecho en el gráfico **Temperature History** y seleccione **Visible Items»Scale Legend** en el menú contextual.

   - ❑ Haga clic con el botón derecho en el gráfico **Temperature History** y seleccione **Visible Items»Graph Palette** en el menú contextual.

7. Guarde el VI.

Prueba

1. Ejecute el VI. Use las herramientas de la leyenda de escalas y la paleta de gráfico para examinar los datos que genere.

2. Cambie el interruptor **Power** a la posición **Off** para detener el VI.

3. Cierre el VI tras terminar.

Fin del ejercicio 4-5

I. Estructuras Case

Una estructura Case, como la de abajo, tiene dos o más subdiagramas o casos.

Sólo se ve un subdiagrama a la vez y la estructura ejecuta sólo un caso cada vez. Un valor de entrada determina qué subdiagrama ejecutar. La estructura Case es similar a las declaraciones switch o a las declaraciones if...then...else en lenguajes de programación basados en texto.

La etiqueta del selector de caso en la parte superior de la estructura Case, mostrada abajo, contiene en el centro el nombre del valor de selección que se corresponde con el caso actual y las flechas de incremento y decremento a cada lado.

Haga clic en las flechas de incremento y decremento para desplazarse por los casos disponibles. También puede hacer clic en la flecha junto al nombre del caso y seleccionar un caso del menú desplegable.

Cablee un valor de entrada, o selector, al terminal de selección, mostrado abajo, para determinar qué caso se ejecuta.

Debe cablear un número entero, un valor booleano, una cadena de caracteres o un valor de tipo enumerado en el terminal de selección. Puede colocar el terminal de selección en cualquier parte del borde izquierdo de la estructura Case. Si el tipo de dato del terminal de selección es booleano, la estructura tiene un caso `TRUE` y uno `FALSE`. Si el terminal de selección es un número entero, cadena de caracteres o valor de tipo enumerado, la estructura puede tener cualquier número de casos.

Si no especifica un caso predeterminado para que la estructura Case controle valores fuera de rango, debe enumerar explícitamente cada valor de entrada posible. Por ejemplo, si el selector es un número entero y especifica casos para `1`, `2` y `3`, debe especificar un caso predeterminado a ejecutar si el valor de entrada es `4` u otro valor de número entero sin especificar.

Nota No puede especificar un caso predeterminado si cablea un control booleano al selector. Si hace clic con el botón derecho en la etiqueta del selector de caso, **Make This The Default Case** no aparecerá en el menú contextual. Seleccione el valor TRUE o FALSE en el control booleano para determinar qué caso ejecutar.

Para convertir una estructura Case en una estructura Stacked Sequence, haga clic con el botón derecho en la estructura Case Sequence y seleccione **Replace with Stacked Sequence** en el menú contextual.

Haga clic con el botón derecho en el borde de la estructura Case para añadir, duplicar, eliminar o reordenar casos y para seleccionar un caso predeterminado.

Seleccionar un caso

La figura 4-40 muestra un VI que utiliza una estructura Case para ejecutar código diferente en función de si un usuario selecciona °C o °F como unidades de temperatura. El diagrama de bloques superior muestra el caso True en primer plano. En el diagrama de bloques intermedio, se selecciona el caso False. Para seleccionar un caso, escriba el valor en el identificador del selector de caso o use la herramienta de etiquetado para modificar los valores. Tras seleccionar otro caso, ése aparecerá en el diagrama de bloques, como se muestra en el diagrama de bloques inferior de la figura 4-40.

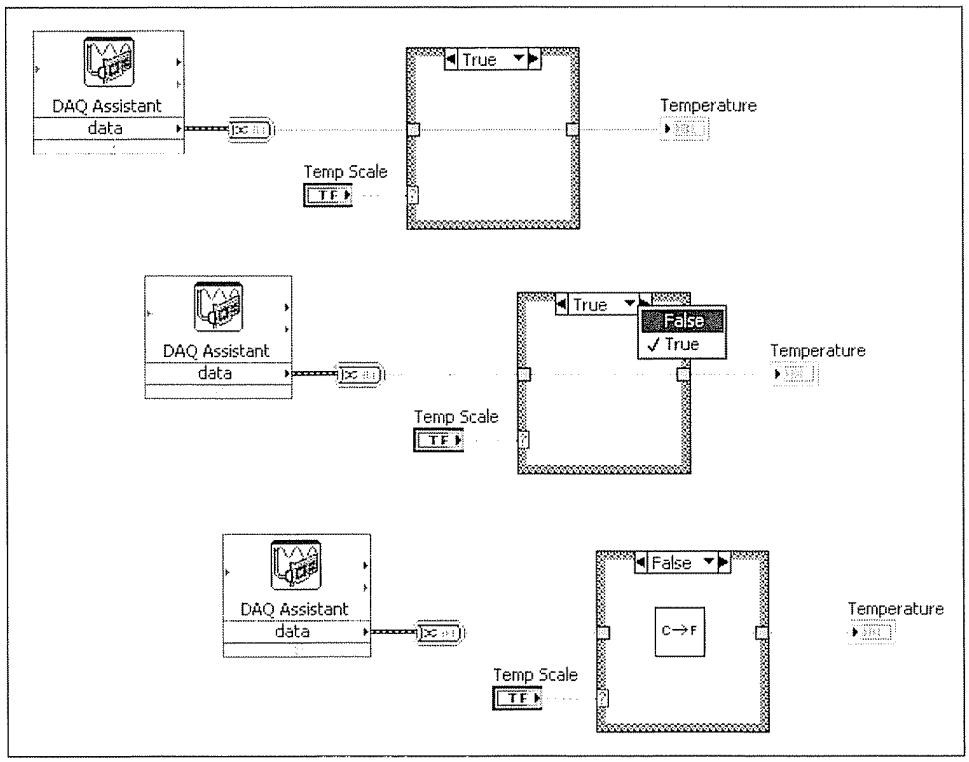


Figura 4-40. Cambiar la visualización de caso de una estructura Case

Si introduce un valor selector que no es del mismo tipo que el objeto cableado al terminal de selección, el valor aparece en rojo. Esto indica que el VI no se ejecutará hasta que elimine o modifique el valor. Asimismo, debido al posible error de redondeo inherente de la aritmética de coma flotante, no puede utilizar números en coma flotante como valores de

selección de caso. Si cablea un valor en coma flotante al caso, LabVIEW redondeará el valor al número entero par más cercano. Si escribe un valor en coma flotante en la etiqueta del selector de caso, el valor aparecerá rojo para indicar que debe eliminar o modificar el valor antes de que pueda ejecutarse la estructura.

Túneles de entrada y salida

Puede crear varios túneles de entrada y salida para una estructura Case. Las entradas están disponibles en todos los casos, pero los casos no necesitan utilizar cada entrada. No obstante, debe definir un túnel de salida para cada caso.

Piense en este ejemplo: una estructura Case en el diagrama de bloques tiene un túnel de salida, pero en al menos uno de los casos no hay valor de salida cableado al túnel. Si ejecuta este caso, LabVIEW no sabrá qué valor devolver para la salida. LabVIEW indica este error dejando el centro del túnel en blanco. El caso sin cablear podría no ser el caso que se está viendo en el diagrama de bloques.

Para corregir este error, muestre los casos que contienen el túnel de salida sin cablear y cablee una salida al túnel. También puede hacer clic con el botón derecho en el túnel de salida y seleccionar **Use Default If Unwired** en el menú contextual para utilizar el valor predeterminado para el tipo de datos del túnel en todos los túneles sin cablear. Cuando se cablea la salida en todos los casos, el túnel de salida aparece relleno con el color del tipo de dato asociado al mismo.

Evite usar la opción **Use Default If Unwired**. Si usa esta opción, no documentará bien el diagrama de bloques y podrá confundir a otros programadores que utilizan su código. La opción **Use Default If Unwired** también dificulta la depuración del código. Si usa esta opción, tenga en cuenta que el valor predeterminado utilizado es el del tipo de datos que está conectado al túnel. Por ejemplo, si el túnel es un tipo de datos booleano, el valor predeterminado es `FALSE`. Consulte la tabla 4-3 para ver una lista de valores predeterminados para los tipos de datos.

Tabla 4-3. Valores predeterminados de tipos de datos

Tipo de datos	Valor predeterminado
Numérico	`0`
Booleano	`FALSE`
Cadena de caracteres	vacío (`""`)

Ejemplos

En los siguientes ejemplos, los valores numéricos pasan por túneles a la estructura Case y se suman o restan, en función del valor cableado al terminal de selección.

Estructura Case booleana

La figura 4-41 muestra una estructura Case booleana. Los casos se solapan entre sí para simplificar la ilustración.

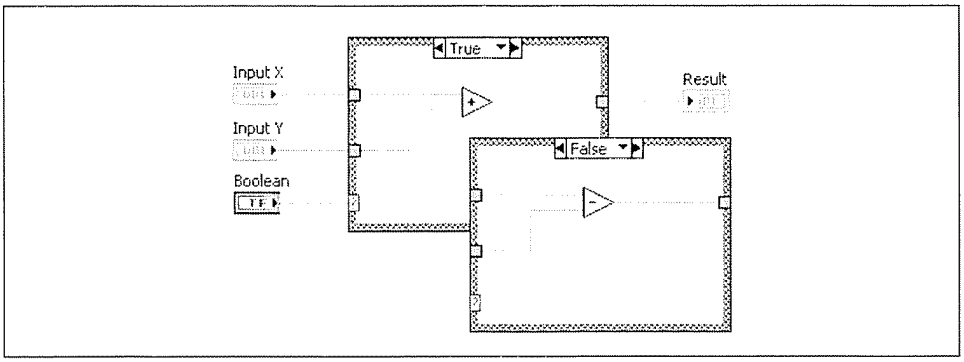


Figura 4-41. Estructura Case booleana

Si el control booleano cableado al terminal de selección es True, el VI suma los valores numéricos. De lo contrario, el VI resta los valores numéricos.

Estructura Case de número entero

La figura 4-42 muestra una estructura Case de número entero.

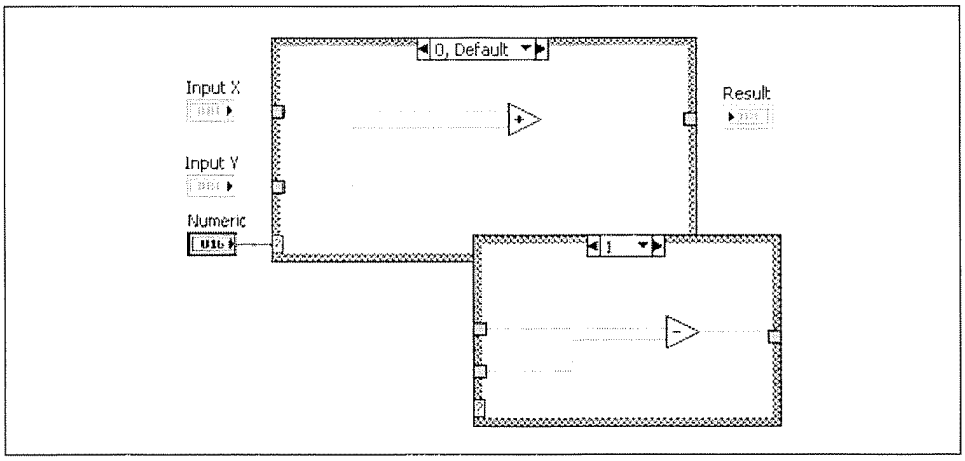


Figura 4-42. Estructura Case de número entero

Numeric es un control de anillo de texto situado en la paleta **Text Controls** que asocia valores numéricos con elementos de texto. Si el número entero cableado al terminal de selección es `0` (`add`), el VI suma los valores numéricos. Si el valor es `1` (`subtract`), el VI resta los valores numéricos.

Si Numeric es cualquier otro valor distinto de `0` (`add`) o `1` (`subtract`), el VI suma los valores numéricos, porque ése es el caso predeterminado.

Estructura Case de cadena de caracteres

La figura 4-43 muestra una estructura Case de cadena de caracteres.

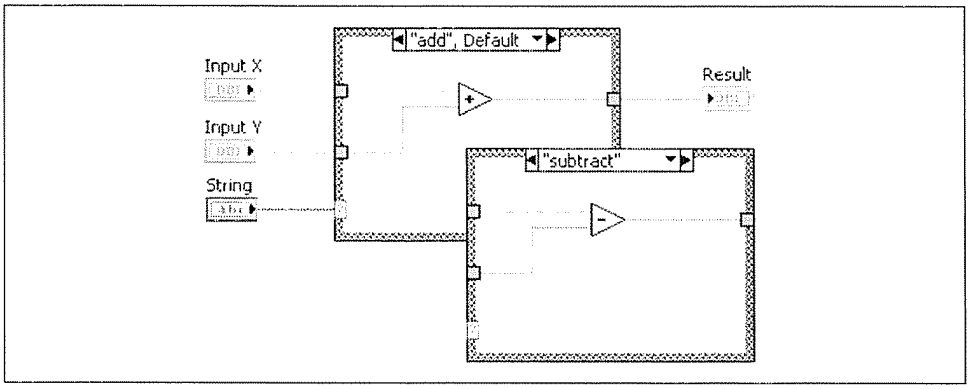


Figura 4-43. Estructura Case de cadena de caracteres

Si **String** es `add`, el VI suma los valores numéricos. Si **String** es `subtract`, el VI resta los valores numéricos.

Estructura Case enumerada

La figura 4-44 muestra una estructura Case enumerada.

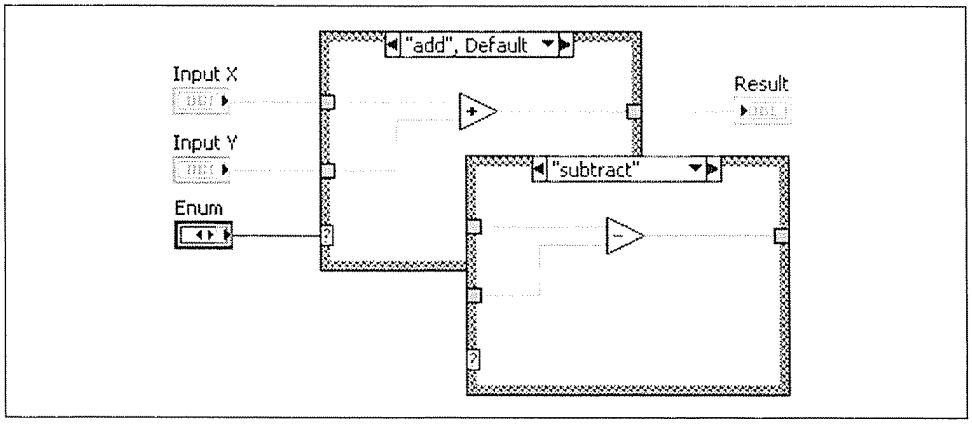


Figura 4-44. Estructura Case enumerada

Un control de tipo enumerado facilita a los usuarios una lista de elementos para seleccionar. El tipo de datos de un control de tipo enumerado incluye información sobre los valores numéricos y etiquetas de cadenas de caracteres del control. El selector de caso muestra la etiqueta de la cadena de caracteres para cada elemento del control de tipo enumerado cuando selecciona **Add Case For Every Value** en el menú contextual de la estructura Case. La estructura Case ejecuta el subdiagrama de caso

apropiado en función del elemento actual del control de tipo enumerado. En el diagrama de bloques anterior, si **Enum** es `add`, el VI suma los valores numéricos. Si **Enum** es `subtract`, el VI resta los valores numéricos.

Uso de estructuras Case para gestionar errores

El siguiente ejemplo muestra una estructura Case donde un cluster de error define los casos.

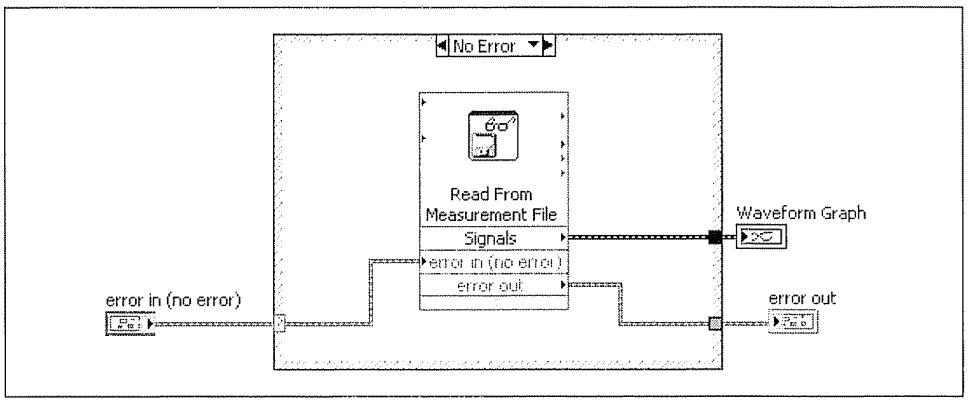


Figura 4-45. Caso No Error

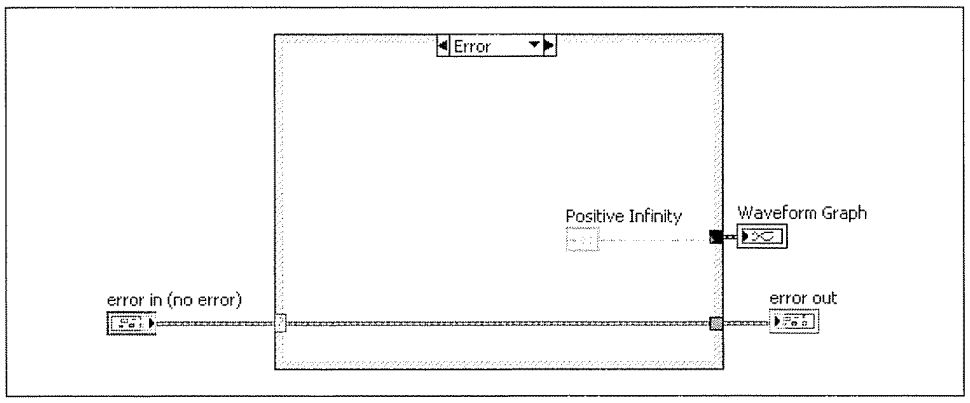


Figura 4-46. Caso Error

Cuando cablea un cluster de error al terminal de selección de una estructura Case, la etiqueta del selector de caso muestra dos casos: `Error` y `No Error`, y el borde de la estructura Case cambia de color: rojo para `Error` y verde para `No Error`. Si ocurre un error, la estructura Case ejecuta el subdiagrama `Error`.

Cuando cablea un clúster de error al terminal de selección, la estructura Case sólo reconoce el elemento booleano **status** del cluster.

Ejercicio 4-6 VI Determine Warnings

Objetivo

Modificar un VI para utilizar una estructura Case y tomar una decisión software.

Escenario

Ha creado un VI en el que el usuario introduce una temperatura, una temperatura máxima y una mínima. Se genera una cadena de caracteres de advertencia en función de la relación de las entradas. No obstante, una situación podría hacer que el VI funcionara incorrectamente. El usuario podría introducir una temperatura máxima inferior a la temperatura mínima. Modifique el VI para generar otra cadena de caracteres que alerte al usuario del: "`Upper Limit < Lower Limit`." Configure el indicador **Warning?** en True para indicar el error.

Diseño

Modifique el diagrama de flujo creado para el VI Determine Warnings original como en la figura 4-47.

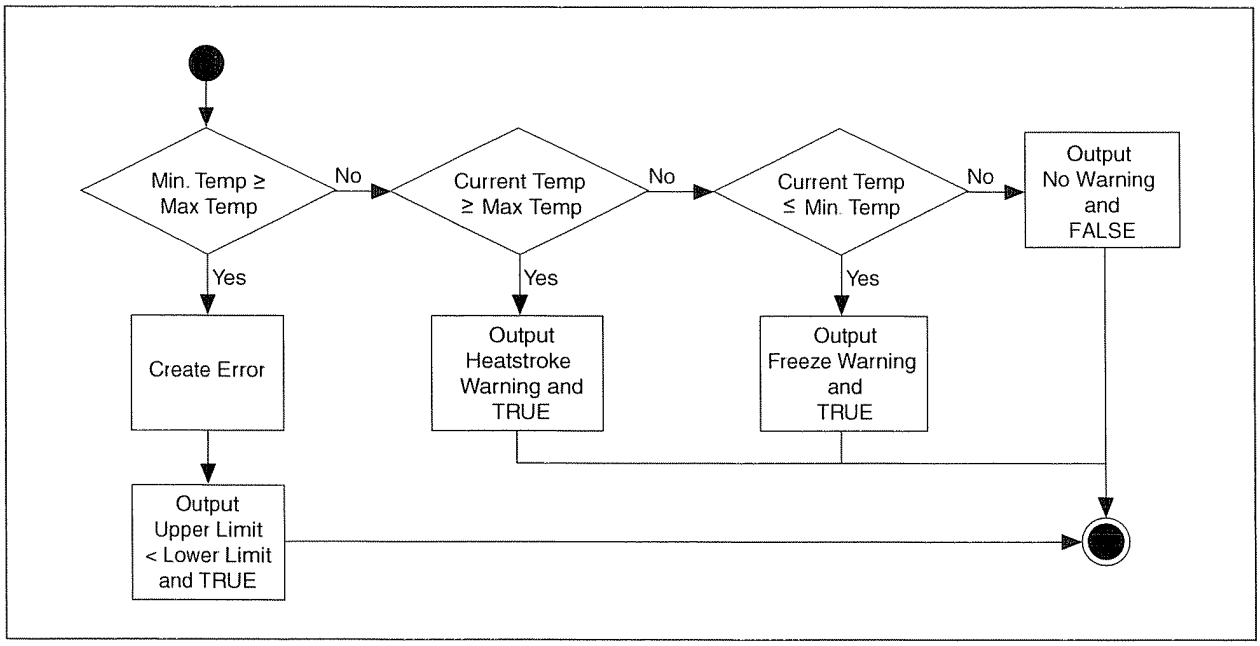


Figura 4-47. Diagrama de flujo de Determine Warnings modificado

El diagrama de bloques original del VI Determine Warnings aparece en la figura 4-48. Este VI debe tener una estructura Case añadida para ejecutar el código si la temperatura máxima es mayor o igual que la temperatura mínima. De lo contrario, el código no se ejecutará. En cambio, se genera una nueva cadena de caracteres y el indicador **Warning?** se establece a True.

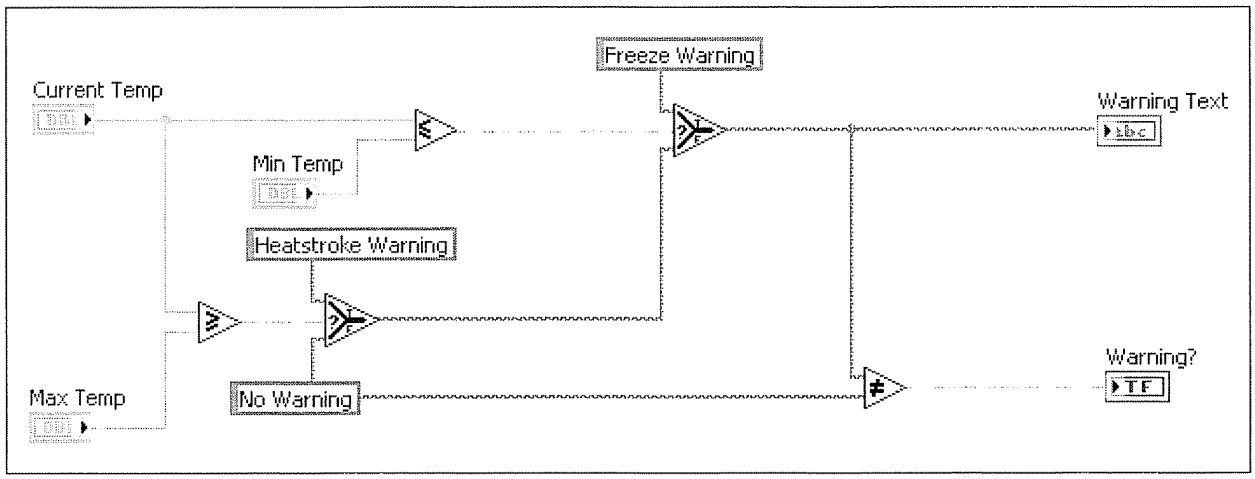


Figura 4-48. Diagrama de bloques del VI Determine Warnings

Implementación

Siga las instrucciones inferiores para crear un diagrama de bloques similar al de la figura 4-49. Este VI forma parte del proyecto de la estación meteorológica de temperatura.

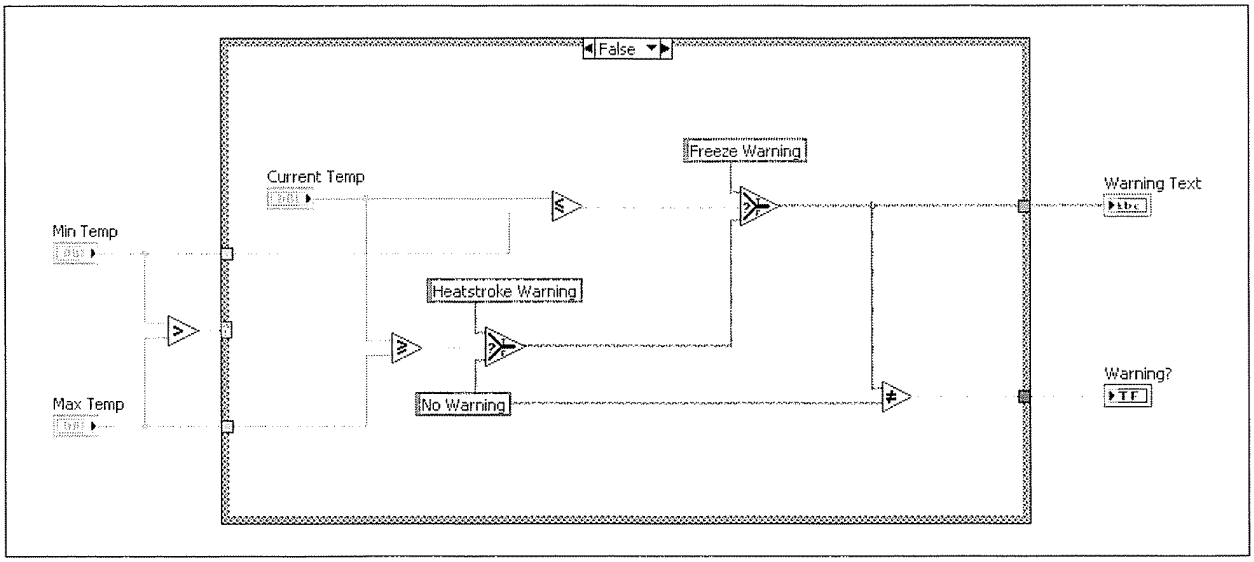


Figura 4-49. Diagrama de bloques del VI Determine Warnings

1. Abra el VI Determine Warnings en el directorio `<Exercises>\LabVIEW Basics I\Determine Warnings`.

2. Abra el diagrama de bloques.

3. Cree espacio en el diagrama de bloques para añadir la estructura Case.

   Los controles **Max Temp** y **Min Temp** y los indicadores **Warning Text** y **Warning?** deben estar fuera de la nueva estructura Case, porque ambos casos de la estructura Case utilizan estos indicadores y controles.

   - ❑ Seleccione los terminales de control **Min Temp** y **Max Temp**.

Consejo Para seleccionar más de un elemento, pulse la tecla <Shift> mientras selecciona los elementos.

   - ❑ Mientras los terminales siguen seleccionados, use la tecla de flecha izquierda del teclado para mover los controles a la izquierda.

Consejo Mantenga pulsada la tecla <Mayús> para mover los objetos en incrementos de cinco píxeles.

   - ❑ Seleccione los terminales indicadores **Warning Text** y **Warning?**.
   - ❑ Alinee los terminales seleccionando **Align Objects»Left Edges**.
   - ❑ Mientras los terminales siguen seleccionados, use la tecla de flecha derecha del teclado para mover los indicadores a la derecha.

4. Compare **Min Temp** y **Max Temp**.

   - ❑ Añada la función Greater? al diagrama de bloques.
   - ❑ Cablee la salida **Min Temp** a la entrada **x** de la función Greater?.
   - ❑ Cablee la salida **Max Temp** a la entrada **y** de la función Greater?.
   - ❑ Añada una estructura Case alrededor del código del diagrama de bloques, salvo para los terminales excluidos.
   - ❑ Cablee la salida de la función Greater? al terminal de selección de caso de la estructura Case.

5. Si **Min Temp** es inferior a **Max Temp**, ejecute el código que determina el indicador y la cadena de caracteres de advertencia.

   - ❑ Mientras permanece visible el caso True, haga clic con el botón derecho en la estructura Case y seleccione **Make This Case False** en el menú contextual. Cuando cree una estructura Case alrededor del código existente, éste se colocará automáticamente en el caso True.

6. Si **Min Temp** es mayor que **Max Temp**, cree una cadena de caracteres personalizada para el indicador **Warning Text** y configure el indicador **Warning?** en True, como muestra la figura 4-50.

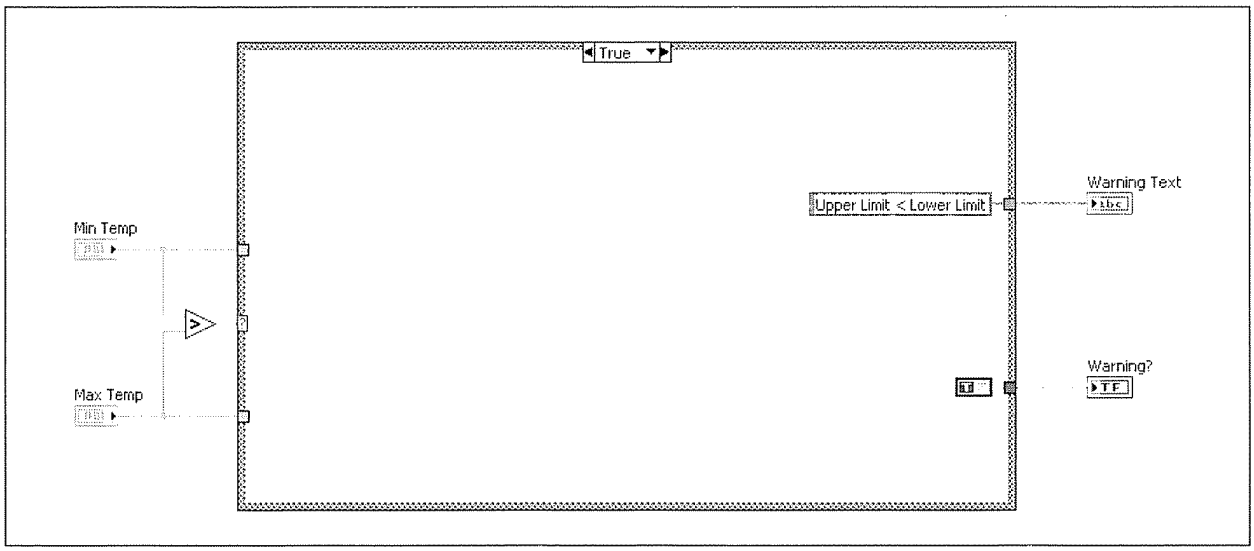


Figura 4-50. Diagrama de bloques del VI Determine Warnings

- ❑ Seleccione el caso **True**.
- ❑ Haga clic con el botón derecho en el túnel de salida de cadena de caracteres.
- ❑ Seleccione **Create»Constant**.
- ❑ Escriba `Upper Limit < Lower Limit` en la constante.
- ❑ Haga clic con el botón derecho en el túnel de salida Warning?.
- ❑ Seleccione **Create»Constant**.
- ❑ Use la herramienta de operaciones para cambiar la constante a una constante True.

7. Guarde el VI.

Prueba

1. Cambie al panel frontal del VI.
2. Cambie el tamaño del indicador **Warning Text** para que pueda contener la nueva cadena de caracteres.
3. Pruebe el VI introduciendo valores de la tabla 4-4 para **Current Temp**, **Max Temp** y **Min Temp** y ejecutando el VI para cada conjunto de datos.

   La tabla 4-4 muestra los valores esperados de Warning Text y Warning? para cada conjunto de datos.

Tabla 4-4. Valores de prueba para Determine Warnings.vi

Current Temp	Max Temp	Min Temp	Warning Text	Warning?
30	30	10	**Heatstroke Warning**	True
25	30	10	**No Warning**	False
10	30	10	**Freeze Warning**	True
25	20	30	**Upper Limit < Lower Limit**	True

4. Guarde y cierre el VI.

Fin del ejercicio 4-6

Ejercicio 4-7 Autoestudio: VI Square Root

Objetivo

Crear un VI que utilice una estructura Case para tomar una decisión de software.

Escenario

Crear un VI que calcule la raíz cuadrada de un número que introduzca el usuario. Si el número es negativo, muestra el siguiente mensaje al usuario: `Error...Negative Number`.

Diseño

Entradas y salidas

Tabla 4-5. Entradas y salidas

Tipo	Nombre	Propiedades
Entrada	`Number`	Doble precisión y coma flotante; valor predeterminado de 25
Salida	`Square Root Value`	Doble precisión y coma flotante

Diagrama de flujo

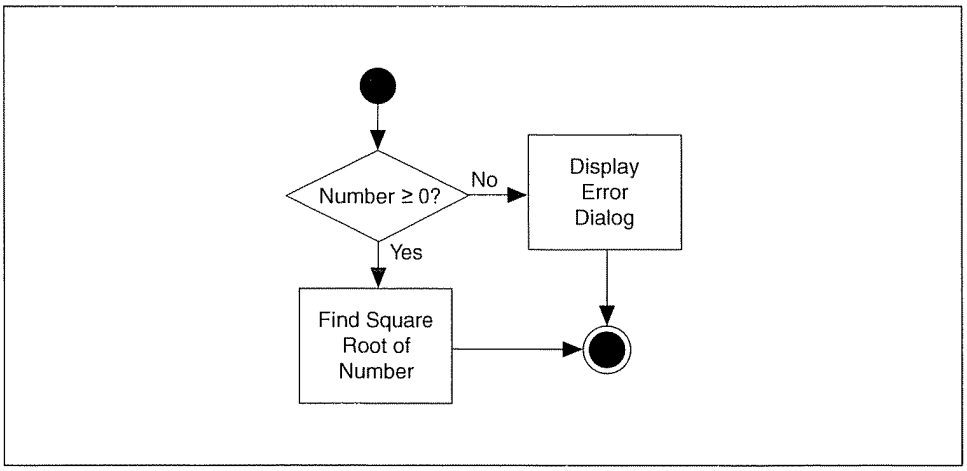


Figura 4-51. Diagrama de flujo del VI Square Root

Implementación

1. Abra un VI nuevo y cree el panel frontal de la figura 4-52.

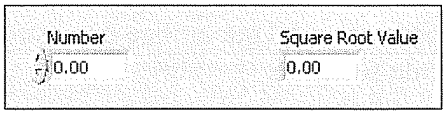


Figura 4-52. Panel frontal del VI Square Root

2. Añada un control numérico a la ventana del panel frontal.
   - ❑ Llame al control numérico `Number`.
3. Añada un indicador numérico a la ventana del panel frontal.
   - ❑ Llame al indicador numérico `Square Root Value`.

Cree el diagrama de bloques de la figura 4-53.

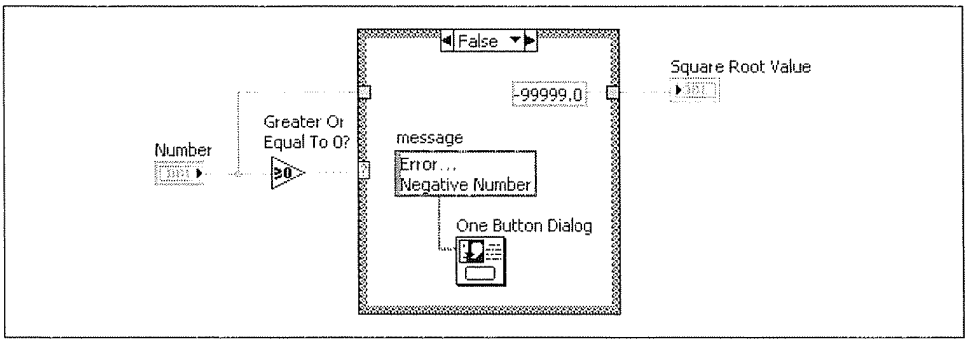
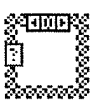


Figura 4-53. Diagrama de bloques del VI Square Root

4. Determine si **Number** es mayor o igual que cero, porque no puede calcular la raíz cuadrada de un número negativo.

   - ❑ Añada la función Greater or Equal to 0? a la derecha del control **Number**. Esta función devuelve True si **Number** es mayor o igual que 0.
   - ❑ Cablee **Number** a la entrada de la función Greater or Equal to 0?.
5. Si **Number** es menor que 0, muestre un cuadro de diálogo que informe al usuario del error.



   - ❑ Añada la estructura Case al diagrama de bloques.
   - ❑ Haga clic en el botón de disminuir o aumentar para seleccionar el caso False.

- ❑ Añada una constante numérica al caso False.
- ❑ Haga clic con el botón derecho en la constante numérica y seleccione **Representation»DBL**.
- ❑ Escriba `-99999` en la constante numérica.
- ❑ Cablee la constante numérica al borde derecho de la estructura Case.
- ❑ Cablee el nuevo túnel al indicador **Square Root Value**.

- ❑ Añada la función One Button Dialog al caso False. Esta función muestra un cuadro de diálogo que contiene un mensaje que especifique.

- ❑ Haga clic con el botón derecho en la entrada del mensaje de la función One Button Dialog y seleccione **Create»Constant** en el menú contextual.
- ❑ Escriba `Error...Negative Number` en la constante.
- ❑ Termine de cablear el caso False como en la figura 4-53.

6. Si **Number** es mayor o igual que 0, calcule la raíz cuadrada del número.

- ❑ Seleccione el caso True de la estructura Case.
- ❑ Coloque la función Square Root en el caso True. Esta función devuelve la raíz cuadrada de **Number**.
- ❑ Cablee la función como en la figura 4-54.

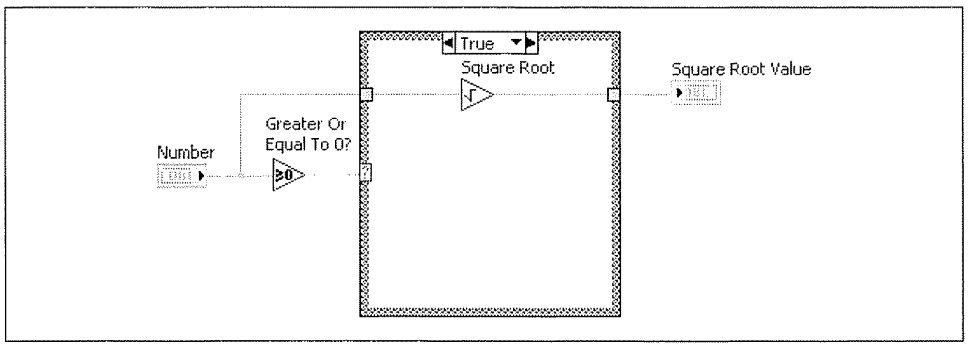


Figura 4-54. Caso True del VI Square Root

7. Guarde el VI como `Square Root.vi` en el directorio `<Exercises>\LabVIEW Basics I\Square Root`.

Prueba

1. Abra el panel frontal.
2. Escriba un número positivo en el control **Number**.
3. Ejecute el VI.
4. Escriba un número negativo en el control **Number**.

Precaución *No* ejecute este VI continuamente. Bajo ciertas circunstancias, ejecutar continuamente este VI puede provocar un bucle infinito.

5. Ejecute el VI.

   Si **Number** es positivo, el VI ejecuta el caso True y devuelve la raíz cuadrada de **Number**. Si **Number** es negativo, el VI ejecuta el caso False, devuelve `-99999` y muestra un cuadro de diálogo con el mensaje `Error...Negative Number`.

6. Cierre el VI.

Fin del ejercicio 4-7

Ejercicio 4-8 Autoestudio: VI Determine Warnings (Reto)

Objetivo

Modificar un VI existente para utilizar una estructura Case o el Formula Node y tomar una decisión software.

Escenario

En el VI Determine Warnings del ejercicio 4-1, seleccionó la función Select para pasar una cadena de caracteres basada en una decisión. Revise este diagrama de bloques para usar una estructura Case o Formula Node (o una combinación de ambos) para completar el mismo objetivo.

Diseño

Entradas y salidas

Tabla 4-6. Entradas y salidas del VI Determine Warnings

Tipo	Nombre	Propiedades
Control numérico	`Current Temp`	Doble precisión y coma flotante
Control numérico	`Max Temp`	Doble precisión y coma flotante
Control numérico	`Min Temp`	Doble precisión y coma flotante
Indicador de cadena de caracteres	`Warning Text`	Tres posibles valores: Heatstroke Warning, No Warning o Freeze Warning
LED redondo	`Warning?`	—

Diagrama de flujo

La figura 4-55 muestra el diagrama de flujo que utilizó en el ejercicio 4-1 para crear el VI Determine Warnings.

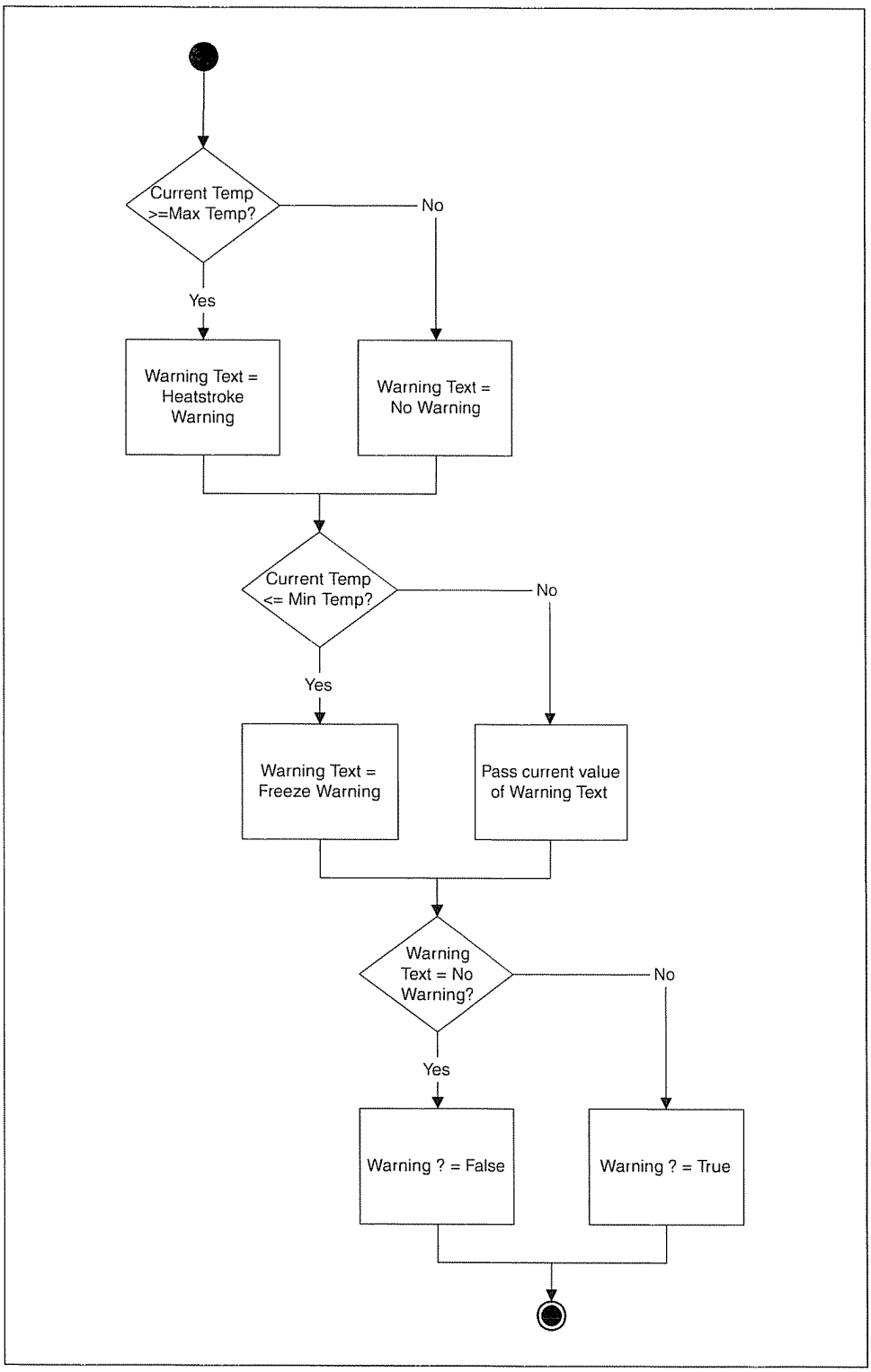


Figura 4-55. Diagrama de flujo del VI Determine Warnings

Implementación

Como parte del reto, no se incluyen instrucciones de implementación en este ejercicio. Abra el VI en el directorio `<Exercises>\LabVIEW Basics I\Determine Warnings Challenge`.

Si necesita ayuda, abra los VIs de soluciones, que se encuentran en el directorio `<Solutions>\LabVIEW Basics I\Exercise 4-8`.

Fin del ejercicio 4-8

Ejercicio 4-9 Autoestudio: VI Determine More Warnings

Objetivo

Manipular cadenas de caracteres con funciones String.

Escenario

Tiene un VI que determina si ha ocurrido una advertencia de golpe de calor o una advertencia de congelación, en función de la entrada de temperatura. Debe ampliar este VI para que también determine si ha ocurrido una advertencia de fuerte viento en función de una lectura de la velocidad del viento y un ajuste de velocidad máxima del viento. Las advertencias deben mostrarse como una única cadena de caracteres. Por ejemplo, si ha ocurrido una advertencia de golpe de calor y otra de fuerte viento, la cadena de caracteres mostrará: Heatstroke and High Wind Warning (advertencia de golpe de calor y de fuerte viento).

Diseño

Entradas y salidas

Tabla 4-7. Entradas y salidas del VI Determine More Warnings

Tipo	Nombre	Propiedades
Control numérico	`Current Temp`	Doble precisión y coma flotante
Control numérico	`Max Temp`	Doble precisión y coma flotante
Control numérico	`Temperatura mín.`	Doble precisión y coma flotante
Control numérico	`Current Wind Speed`	Doble precisión y coma flotante
Control numérico	`Max Wind Speed`	Doble precisión y coma flotante
Indicador de cadena de caracteres	`Warning Text`	Valores posibles: Heatstroke Warning, Freeze Warning, Heatstroke and High Wind Warning, Freeze and High Wind Warning, High Wind Warning y No Warning
Indicador booleano	`Warning?`	Booleano

Diagrama de flujo

El diagrama de flujo de la figura 4-56 se usa para el VI Determine Warnings. Este VI no toma datos del viento. Modifique este diagrama de flujo para determinar también la advertencia de fuerte viento.

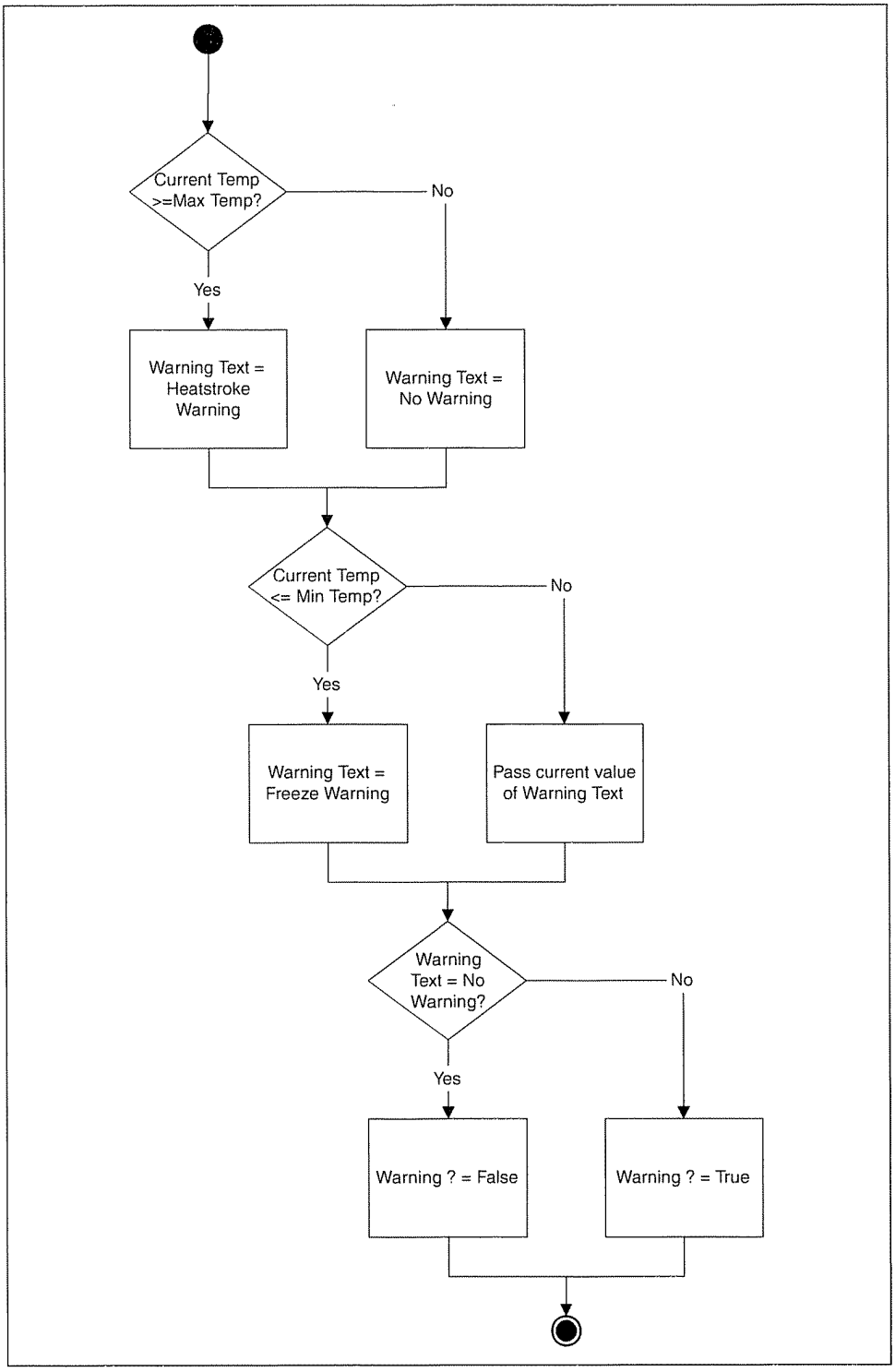


Figura 4-56. Diagrama de flujo del VI Determine Warnings

El diagrama de flujo modificado de la figura 4-57 determina la advertencia de fuerte viento además de las advertencias ya determinadas.

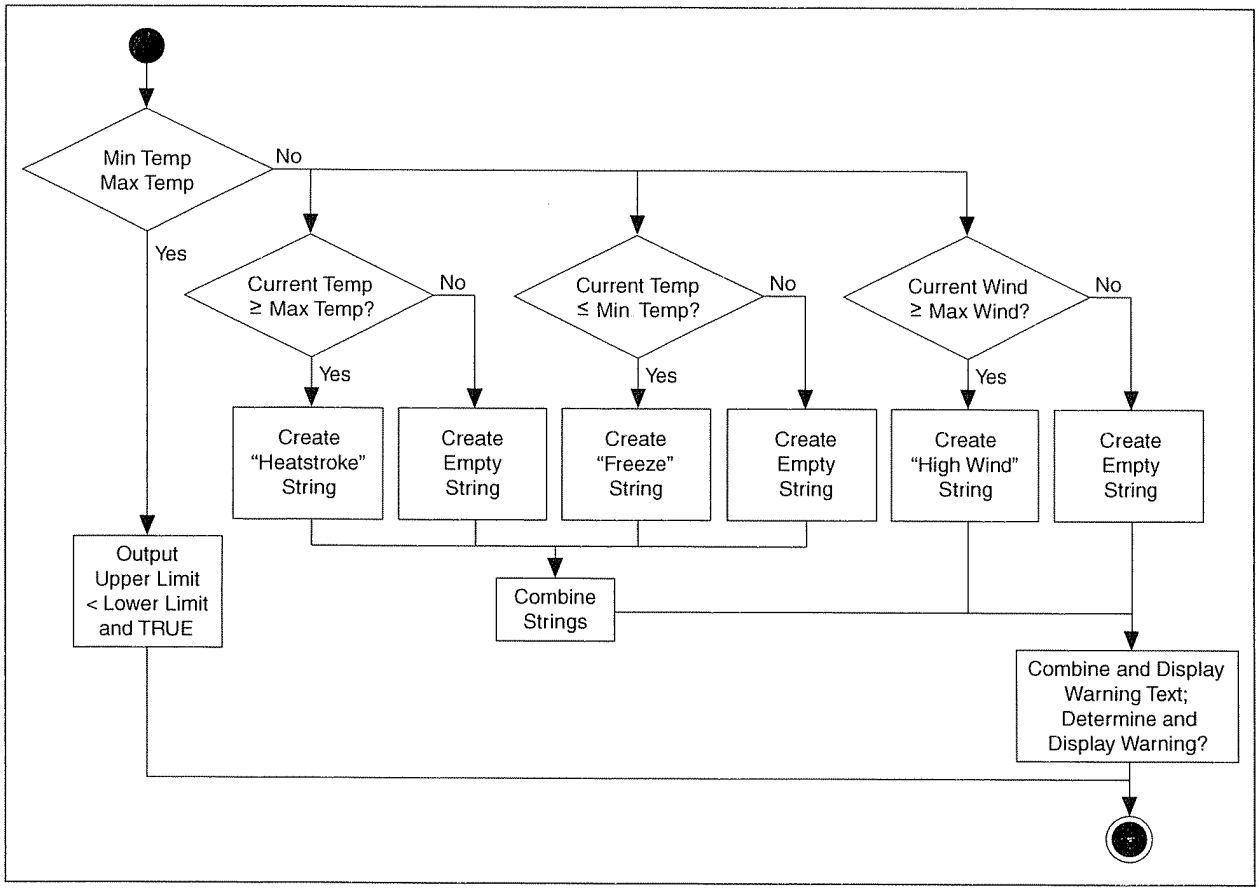


Figura 4-57. Diagrama de flujo del VI Determine More Warnings

Arquitectura del VI

Existen muchos modos de escribir este VI. En este ejercicio utilizará estructuras Case para determinar qué cadena de caracteres pasar y la función Concatenate Strings para unir cadenas de caracteres.

Implementación

Una parte de este VI ya se ha creado. El panel frontal del VI aparece en la figura 4-58. Este panel frontal recupera los valores del usuario para la temperatura actual, la máxima, la mínima, la velocidad actual del viento y la velocidad máxima del viento y muestra al usuario la cadena de caracteres de advertencia y el LED de advertencia. El proyecto de la estación meteorológica de este curso no utiliza este VI.

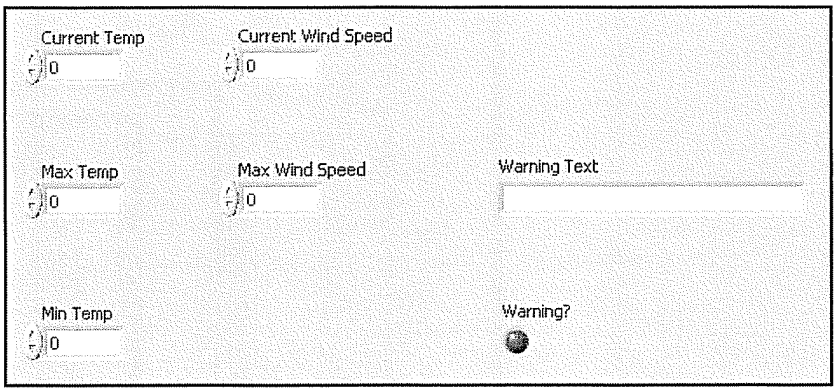


Figura 4-58. Panel frontal de Determine More Warnings

1. Abra el VI `Determine More Warnings.vi` en el directorio `<Exercises>\LabVIEW Basics I\Determine More Warnings`.

Cree el diagrama de bloques para que se asemeje al de la figura 4-59.

2. Abra el diagrama de bloques.
3. Use las figuras 4-59 a la 4-63 para ayudarle a crear el código del diagrama de bloques.
4. Usará los siguientes objetos del diagrama de bloques en este ejercicio:

- ❑ Estructura Case
- ❑ Constante Empty String
- ❑ Constante Space
- ❑ Función Equal?
- ❑ Función Concatenate Strings

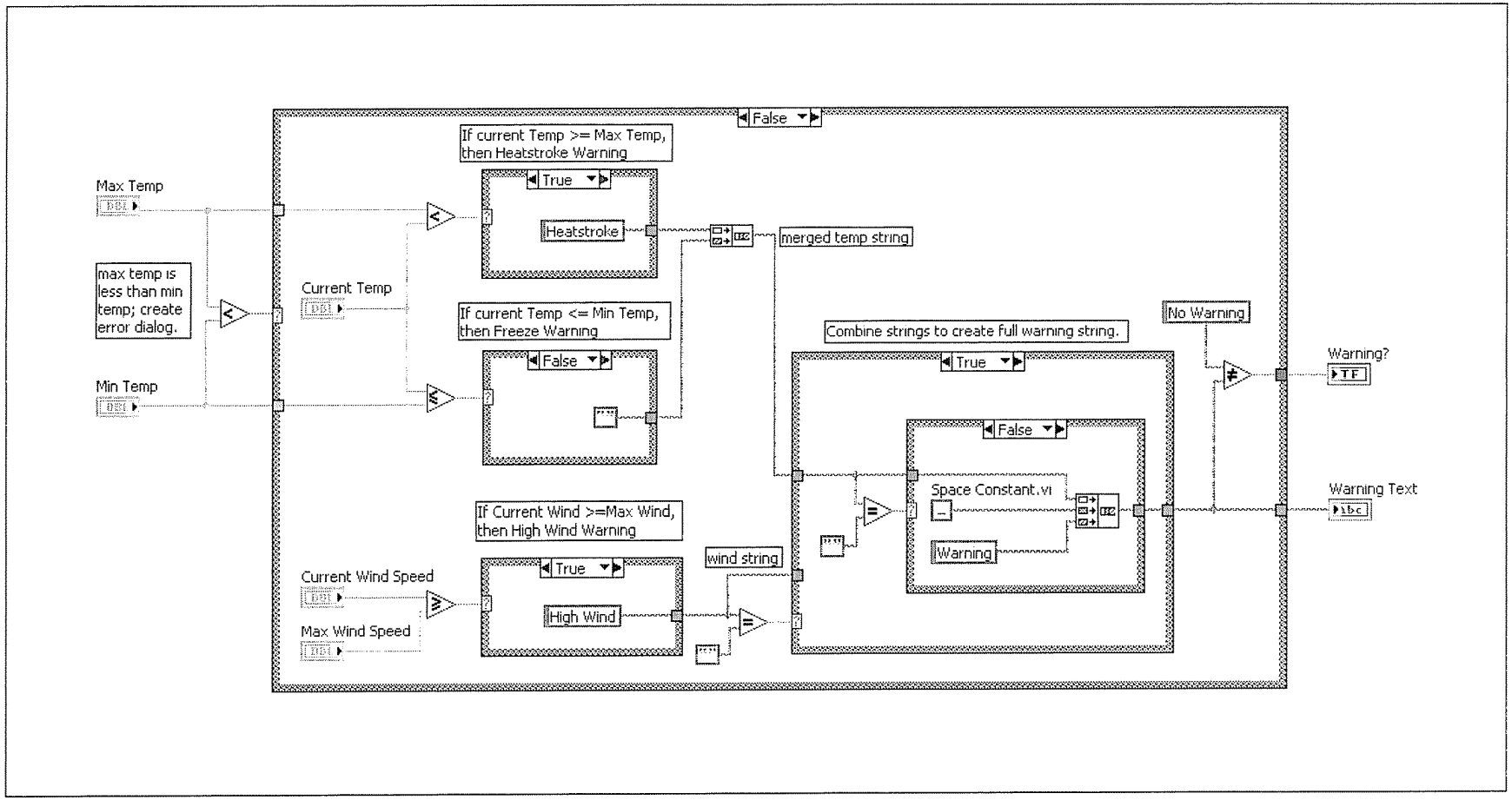


Figura 4-59. Diagrama de bloques de Determine More Warnings

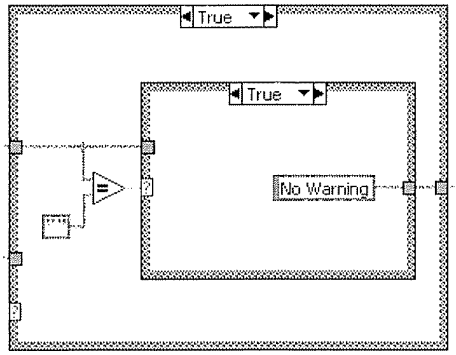


Figura 4-60. Casos True para cuando no se generan las advertencias de temperatura y de viento

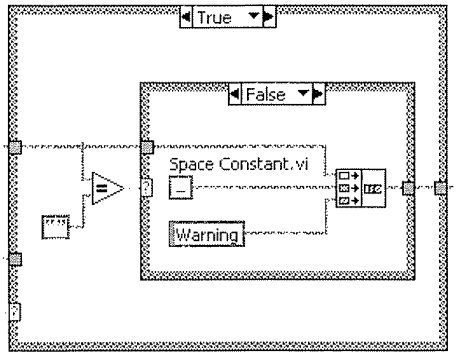


Figura 4-61. Caso True para cuando se genera una advertencia de temperatura

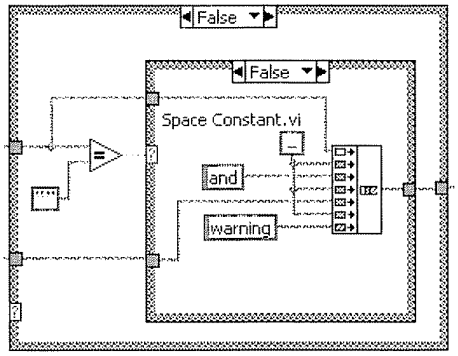


Figura 4-62. Casos False para cuando se generan las advertencias de temperatura y de viento

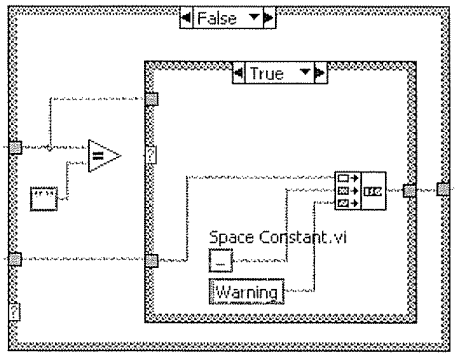


Figura 4-63. Caso False para cuando se genera una advertencia de viento

5. Guarde el VI.

Prueba

1. Pruebe los siguientes valores para asegurarse de que el VI funciona según lo esperado.

Tabla 4-8. Valores de la prueba meteorológica

Nombre	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Prueba 6
Current Temp	20	30	10	30	10	20
Max Temp	25	25	25	25	25	25
Min Temp	15	15	15	15	15	15
Current Wind Speed	25	25	25	35	35	35
Max Wind Speed	30	30	30	30	30	30
Warning Text	No Warning	Heatstroke Warning	Freeze Warning	Heatstroke and High Wind Warning	Freeze and High Wind Warning	High Wind Warning
Warning?	False	True	True	True	True	True

2. Cierre y guarde el VI tras terminar.

Fin del ejercicio 4-9

Autorrevisión: cuestionario

1. ¿Qué elemento de los siguientes identifica el control o indicador en el diagrama de bloques?
   a. Título
   b. Ubicación
   c. Etiqueta
   d. Valor
2. ¿Cuál de las siguientes estructuras deben ejecutarse al menos una vez?
   a. Bucle While
   b. Bucle For
3. ¿Qué elemento de los siguientes SÓLO está disponible en el diagrama de bloques?
   a. Control
   b. Constante
   c. Indicador
   d. Panel de conectores
4. ¿Qué acción mecánica hace que un control booleano en el estado False cambie a True cuando hace clic en él y permanezca True hasta que lo libere y LabVIEW haya leído el valor?
   a. Switch until released
   b. Switch when released
   c. Latch until released
   d. Latch when released

Autorrevisión: respuestas al cuestionario

1. ¿Qué elemento de los siguientes identifica el control o indicador del diagrama de bloques?
   a. Título
   b. Ubicación
   c. Etiqueta
   d. Valor
2. ¿Cuál de las siguientes estructuras deben ejecutarse al menos una vez?
   a. Bucle While
   b. Bucle For
3. ¿Qué elemento de los siguientes SÓLO está disponible en el diagrama de bloques?
   a. Control
   b. Constante
   c. Indicador
   d. Panel de conectores
4. ¿Qué acción mecánica hace que un control booleano en el estado False cambie a True cuando hace clic en él y permanezca True hasta que lo libere y LabVIEW haya leído el valor?
   a. Switch until released
   b. Switch when released
   c. Latch until released
   d. Latch when released