

MINIPROYECTO DE TSIC

PROGRAMARIO BridgeVIEW



Alberto Adrià Escámez
Kenneth Colell Sole
Pedro Alonso Mao
K55

INDICE

	Pagina
1.- Introducción	2
2.- El lenguaje G	4
3. La Paleta de Controles de BridgeVIEW	5
4.- La Paleta de Funciones de BridgeVIEW	14
5.- Aspectos avanzados de BridgeVIEW	19
6.- Creación de una aplicación con BridgeVIEW	26
Bibliografía	31

1.Introducción

BridgeVIEW es un programa SCADA creado por la empresa estadounidense National Instruments, creadora también de los programas LabVIEW y LookOut, y que permite el desarrollo de aplicaciones para la industria de la automatización. Con BridgeVIEW se puede crear un potente interface gráfico para el control y monitorización de cualquier proceso industrial, llamado Human Machine Interface o Interface Hombre Máquina (a partir de este momento nos referiremos a él como HMI) que tenga las siguientes características:

- Fácil de usar.
- Base de datos en tiempo real, a la que nos referiremos como RTDB.
- Crea colecciones de datos históricos relacionados con variables del proceso.
- Informa sobre el estado de alarmas.
- Permite la creación de sistemas de seguridad para controlar el acceso de las personas que pueden manipular la planta.

BridgeVIEW incorpora la posibilidad de intercambiar información con autómatas programables (PLCs) y microcontroladores usando protocolos habituales en las comunicaciones industriales, como MODBUS o FieldBus, gracias a los acuerdos conseguidos con fabricantes como MODBUS, SIEMENS y otras empresas relacionadas con el mundo de la automatización industrial que han llegado a un acuerdo de colaboración formando una fundación llamada OPC.

Comparación entre BridgeVIEW y el resto de software de National Instruments

Con este apartado intentaremos dar una idea general sobre cuales son las principales diferencias y similitudes entre BridgeVIEW y el resto de software de National Instruments, LabVIEW y LookOut, para de esta manera poder conocer para que tipo de aplicaciones esta desarrollado este software y cuando puede ser más útil su utilización.

Comparación entre BridgeVIEW y LabVIEW

Si pudiésemos ver a la vez pantallas para la creación de aplicaciones de BridgeVIEW y LabVIEW nos daríamos cuenta de que, quitando alguna pequeña diferencia, prácticamente son idénticas. Esto es debido a que BridgeVIEW es un programa basado casi en un 100% en LabVIEW, programa dedicado al control de instrumentación, desarrollando para ello aplicaciones mediante el lenguaje de programación gráfico llamado G (y del que hablaremos en la siguiente parte de nuestro trabajo), y que incorpora respecto a este una serie de controladores o drivers para comunicar nuestro PC con autómatas programables, y una serie de nuevas librerías para entre otras cosas crear alarmas que controlen al sistema, gráficas y tablas que recogen la información que vayamos extrayendo de nuestro proceso y un entorno gráfico similar al de la industria real mediante la utilización entre otras cosas de válvulas, tuberías o depósitos.

Es por esta razón por lo que podemos decir que BridgeVIEW no es más que una ampliación de LabVIEW preparada especialmente para trabajar en el mundo de la automatización industrial y que se puede recomendar su utilización a aquellas personas que habiendo utilizado este programa para el desarrollo de aplicaciones en el entorno industrial hayan encontrado algunas limitaciones en este software, ya que LabVIEW no estaba pensado inicialmente para estas aplicaciones y si para aquellas que tengan que ver con el control de instrumentación y el manejo de tarjetas de adquisición de datos. De hecho según afirma la propia empresa National Instruments en sus manuales, aquellas aplicaciones realizadas con LabVIEW podrán ser ejecutadas sin ningún tipo de problema en

BridgeVIEW, mientras que en el caso que queramos realizar la operación contraria, es decir ejecutar una aplicación realizada con BridgeVIEW sobre LabVIEW, sólo será posible en el caso de que no utilicemos herramientas específicas del primer programa.

Comparación entre BridgeVIEW y LookOut

Estos dos programas tienen la misma finalidad: crear aplicaciones para el control, monitorización y supervisión de procesos industriales automatizados, lo que se conoce como SCADA. Esto nos puede llevar a hacernos la pregunta de por qué National Instruments tiene en el mercado dos programas que realizan la misma función. La respuesta la encontramos si vemos la forma en que se crean aplicaciones en uno y otro programa. Mientras que BridgeVIEW se basa en la creación de aplicaciones utilizando el lenguaje G, utilizado también en LabVIEW, LookOut utiliza un sistema gráfico totalmente diferente al empleado hasta ahora, y que no necesita el fundamento mínimo sobre programación necesario para realizar una aplicación en lenguaje G. Así podemos llegar a la conclusión de que la elección de que programa elegiremos para la monitorización y supervisión del proceso vendrá en función de si hemos trabajado con el lenguaje G con anterioridad, en cuyo caso elegiremos BridgeVIEW, y en el caso contrario podremos elegir entre cualquiera de los dos programas, ya que realizan tareas muy parecidas, pero podríamos elegir LookOut si queremos realizar nuestra aplicación de una forma más intuitiva y sencilla.

2.El lenguaje G

G es el lenguaje de programación que se utiliza para desarrollar aplicaciones con BridgeVIEW. Como ya hemos dicho antes en la Introducción este lenguaje es el mismo que se utiliza en LabVIEW por lo que en esta parte no nos dedicaremos a hacer un estudio en profundidad de como se realiza una aplicación con este lenguaje, ya que no es el fin de este trabajo, y además para ello hay un gran número de bibliografía que trata sobre ello, y entre la que podríamos destacar el *G Programming Reference Manual*, realizado por la propia National Instruments, y el *Programación gráfica para el control de instrumentación*, que tiene el añadido de ser el único libro realizado sobre este tema en castellano. Por estas razones nos dedicaremos en esta parte a dar una idea global acerca de lo que es este lenguaje de programación y que partes forman una aplicación realizada con él.

Como BASIC, Pascal o C, G es un lenguaje de propósito general que posee unas extensivas librerías de funciones específicas para por ejemplo la adquisición, análisis, presentación y almacenamiento de datos. Pero lo que diferencia a este lenguaje del resto es que no está basado en texto sino en gráficos, lo que lo convierte en un lenguaje de programación más intuitivo y sencillo de manejar.

Los programas creados en G reciben el nombre de Instrumentos Virtuales (VIs) ya que su apariencia es similar a la de un instrumento real aunque la manera de realizarlos es muy parecida a la de los programas realizados con lenguajes de programación convencionales.

Un VI está formado por un interface interactivo con el usuario, llamado por National Instruments *Front Panel* o Panel Frontal, que es la parte que tiene la apariencia de un instrumento real, y un diagrama que sirve como código fuente del programa, llamado *Block Diagram* o Diagrama de Bloques.

Desde el Panel Frontal el usuario entra la información necesitada por la aplicación y puede visualizar los datos que se vayan extrayendo del proceso en forma de gráficas, números o botones; y el VI recibe instrucciones desde el diagrama de bloques que es la parte que el diseñador de la aplicación debe desarrollar en G.

Una de las mayores ventajas que tiene este lenguaje frente a los lenguajes de programación convencionales es el mejor uso que se hace del concepto de programación modular. Podemos dividir nuestra aplicación en diferentes tareas y crear para cada una de ellas un subVI (como si se tratase de las funciones de los lenguajes convencionales), para finalmente crear un VI principal que englobe a todos los anteriores. No existe un límite sobre los subVI que se pueden usar en un programa en G; y además se puede llamar a un subVI dentro de otro subVI.

3. LA PALETA DE CONTROLES DE BridgeVIEW.

En este apartado, describiremos y pondremos ejemplos visuales de los objetos que nos podemos encontrar en las subpaletas que forman esta Paleta de Controles. Pero sobretodo haremos hincapié en las diferencias o innovaciones que aporta BridgeVIEW respecto a la versión 5.01 de LabVIEW sobre la que nos hemos basado a la hora de calificar estas diferencias.

La Paleta de Controles(*Controls Palette*) solo podremos utilizarla en el Panel Frontal del VI, que será donde pondremos las entradas y salidas del sistema que hayamos diseñado. Podremos visualizar indicadores, gráficas, interruptores, alarmas, leds, etc..

Si al cargar BridgeVIEW vemos que la paleta de funciones no aparece en el Panel Frontal, podemos hacerla visible seleccionando *Windows* y *Show Controls Palette* en el menú del Panel Frontal.

La Paleta de Controles la dividiremos en las siguientes subpaletas (en la siguiente ilustración vemos algunas de estas subpaletas en inglés, para que puedan ser identificadas al seleccionarlas. Posteriormente se explicarán su significado y su traducción):

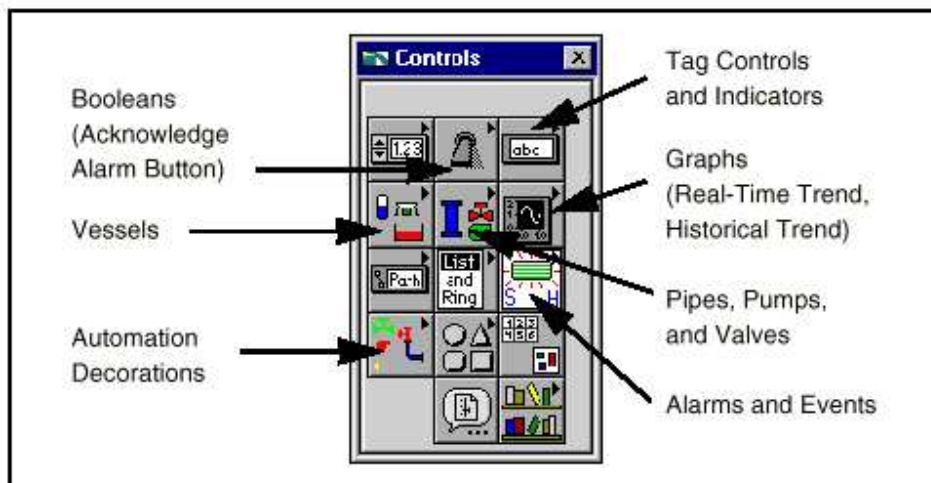


Fig. 1 Formato de la Paleta de Controles.

Describiremos de izquierda a derecha y de arriba a bajo el contenido y daremos algunos ejemplos de los menús desplegables que nos podemos encontrar al seleccionar una de las subpaletas.

NUMERICOS

Usaremos controles numéricos para introducir cantidades numéricas e indicadores para visualizar la información que se produzca en nuestro Diagrama de Bloques. Los controles e indicadores numéricos más utilizados los podemos ver bajo estas líneas, y podremos seleccionarlos en la subpaleta **Numeric** de la Paleta de Controles.

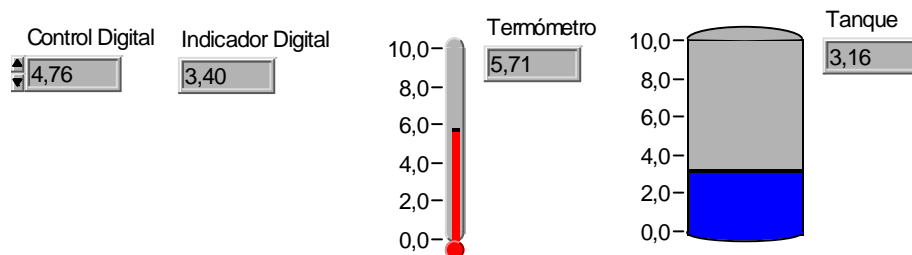


Fig. 2 Indicadores y controles numéricos.

Por ejemplo, mediante estos indicadores podremos visualizar la temperatura de una habitación mediante un termómetro, o ver como se llena o se vacía un tanque.

Respecto a LabVIEW, esta subpaleta no presenta ninguna novedad.

BOOLEANOS.

Utilizaremos controles e indicadores booleanos para valores que puedan resultar CIERTO o FALSO. Los objetos booleanos simulan interruptores, botones y leds. Algunos ejemplos de objetos booleanos los vemos debajo. Estos objetos los encontraremos seleccionando la subpaleta **Boolean**.

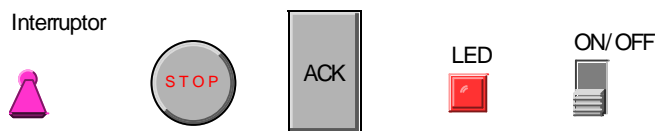


Fig. 3 Indicadores y controles booleanos.

Esta subpaleta respecto a LabVIEW no ofrece prácticamente ninguna novedad, salvo el indicador *ACK*, que podemos ver en la anterior figura y del que hablaremos posteriormente.

STRING.

Usaremos los controles e indicadores *string* para introducir y ver caracteres ASCII. Podemos utilizar *strings* para simples mensajes de texto que provengan de instrumentos o ficheros.

Para seleccionar esta opción deberemos pulsar la subpaleta **String Table Tag**. Si queremos podemos introducir o cambiar el texto del interior del *string*. Si necesitamos minimizar el espacio que el control o indicador *string* ocupa, podemos hacer un *pop-up* (pulsar botón derecho del ratón sobre indicador o control) y seleccionar *Show>>Scrollbar*.

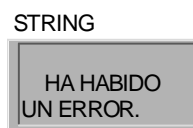


Fig. 4 Indicador String.

Esta subpaleta ofrece las mismas posibilidades que la subpaleta *Strings* de la versión 5.0 de LabVIEW.

VESSELS (Tanques).

En esta subpaleta podemos encontrar varios tipos de tanques. Mediante estos objetos veremos y controlaremos, por ejemplo el nivel de agua de nuestra aplicación.

Este menú desplegable de la Paleta de Controles es innovador en BridgeVIEW respecto a LabVIEW.

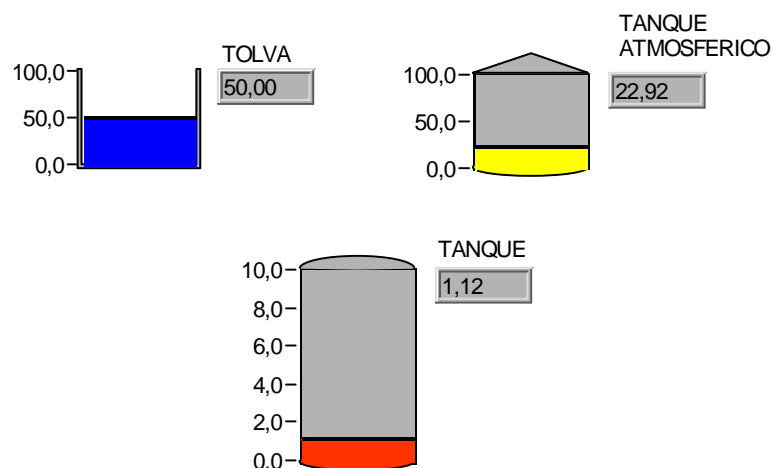


Fig. 5 Indicadores tipo depósitos

PIPES, PUMPS, VALVES.

En esta subpaleta podremos seleccionar todo tipo de tuberías, válvulas y bombas para crear una simulación de nuestra aplicación en el Panel Frontal de nuestro VI. Podemos elegir además la dimensión de estos objetos, pudiendo ser en 2D o 3D.

En la figura siguiente podemos ver algunos ejemplos de estos objetos, que son innovadores en BridgeVIEW.

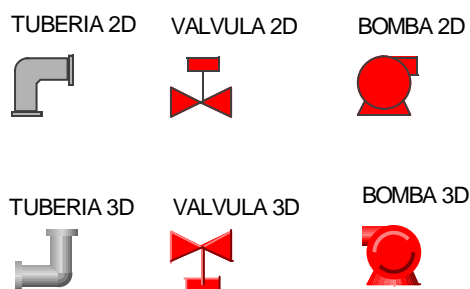


Fig. 6 Pipes, Pumps y Valves.

Este tipo de objetos no son más que controles e indicadores booleanos como los explicados anteriormente.

GRÁFICAS.

Un indicador *graph* o indicador gráfico es una representación bidimensional de una o más gráficas.

Hay tres tipos de indicadores gráficos en la subpaleta **Graph**:

- XY Graph.
- Waveform Graph.
- Intensity graph.

Explicaremos brevemente las diferencias entre estos tres tipos de graphs:

En un *XY Graph*, un punto X puede tener varios valores Y, lo que permite por ejemplo dibujar funciones circulares.

Waveform graph representa una serie de valores Y que están separados un determinado espacio definido por el usuario. Cuando se representa una nueva serie de datos, al contrario de lo que ocurría en los indicadores chart, estos datos se reemplazan a los ya existentes en lugar de añadirse al lado, y pierden los valores representados con anterioridad.

Intensity graph es exactamente igual que *intensity chart* salvo que no retiene valores anteriores, por lo que cuando un nuevo bloque de valores se carga, éstos sustituyen a los ya existentes.

En la figura siguiente vemos un ejemplo de *Waveform Chart*. Esta subpaleta ofrece el mismo tipo de gráficas que la subpaleta de LabVIEW de la Paleta de Controles.

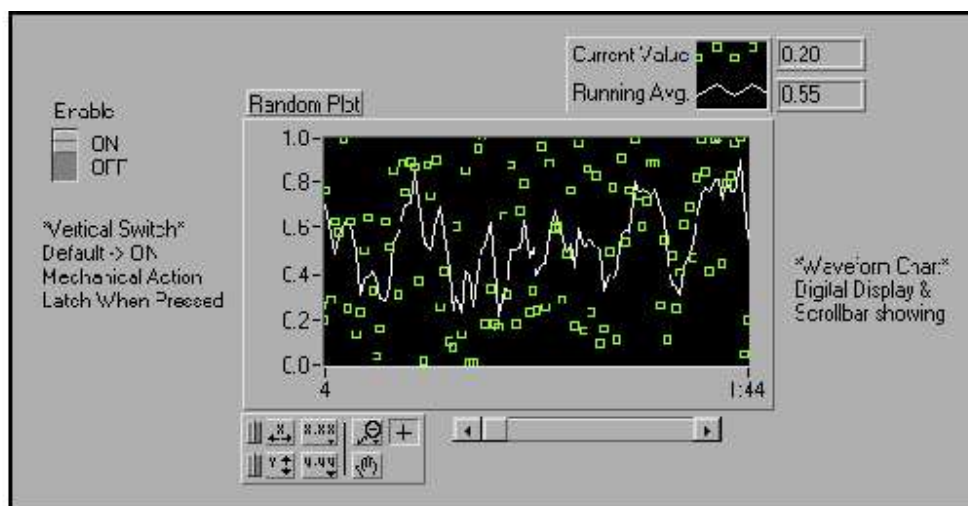


Fig. 7 Ejemplo de Waveform Graph.

Debajo de la representación de la gráfica podemos ver una barra que podemos mover de derecha a izquierda para poder ver ciertos intervalos de la gráfica. Esta opción podemos hacerla visible haciendo un pop-up (botón derecho del mouse) encima de la gráfica y seleccionando *Show>>Scrollbar>>*.

Debajo de la gráfica podemos ver una paleta que nos permitirá cambiar la precisión de los ejes o hacer un zoom de una determinada zona de la gráfica.

Haciendo un *pop-up* encima de la etiqueta *Plot*, podremos cambiar el estilo de línea, el estilo de puntos o el color, entre otras funciones.

ALARMS & EVENTS(Alarmas y acontecimientos).

Mediante esta opción podremos tener una tabla resumen con todos los acontecimientos sobre las alarmas que han sucedido en nuestro VI. Podremos tener una tabla del estado de nuestras alarmas, indicándonos el día y hora, estado de la alarma, si estaba activada o no, su prioridad, etc...

Esta posibilidad es una novedad más de BridgeVIEW, y hablaremos con más detenimiento de ella posteriormente.

Alarm Summary

Date	Time	Tag	Value	Alarm State	Ack Status	Priority

Fig. 8 Indicador Resumen de Alarmas

AUTOMATION OBJECTS (Objetos automatizados).

Mediante esta subpaleta podremos seleccionar tuberías, motores, para insertar decoraciones parecidas a las de la industria real en el Panel Frontal que formará nuestra aplicación.



Fig. 9 Objetos Automatizados.

Esta subpaleta al igual que la de tuberías, bombas y válvulas son nuevas respecto a la versión 5.0 de LabVIEW.

Array & Cluster.

Un *array* es una colección de datos, todos ellos del mismo tipo. La creación de *arrays* de control o indicadores en el Panel Frontal se hace mediante la combinación del icono *array shell* de la subpaleta **Array & Cluster**, con un objeto de datos que puede ser numérico, booleano o *string*.

En cambio, un *Cluster* es una colección ordenada de uno o más elementos. A diferencia de los *arrays*, los *Cluster* pueden contener cualquier combinación de tipos de datos. Otra diferencia es que los clusters tienen un tamaño fijo. Podemos crear un cluster de indicadores o controles poniendo cualquier combinación de booleanos, *strings*, *charts*, *graphs*, escalares, *arrays* o, incluso otros *clusters* dentro de un *cluster shell*.

Como se ve en la figura podemos indexar e introducir elementos dentro de los *arrays* en los bordes de la estructura, en este caso una *FOR* (de la que hablaremos en la siguiente parte del trabajo). En cada iteración dentro del bucle se crea un elemento del array, y una vez completado, el array pasa al indicador.

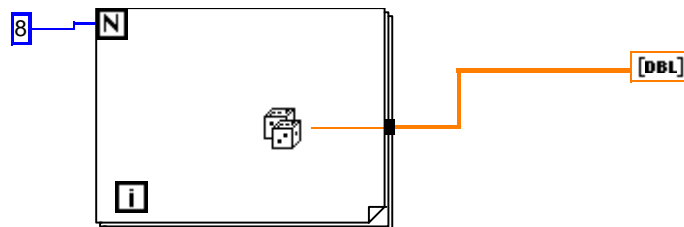
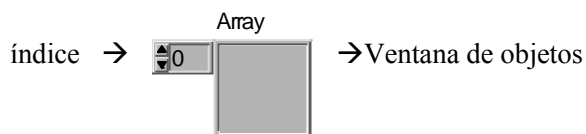


Fig. 10 Creación de un array de números aleatorios

LA PALETA DE HERRAMIENTAS.

La paleta de herramientas la utilizaremos para llevar a cabo funciones específicas de edición o ejecución. Explicaremos su contenido brevemente, ya que no ofrece ninguna novedad respecto a la de LabVIEW.



Fig. 11 Formato de la Tools Palette.



La herramienta *Operating* (Funcionamiento) maneja los controles del panel frontal. Es la única herramienta disponible en el modo *Run*.



La herramienta *Positioning* (Situación) selecciona, mueve y redimensiona objetos.



La herramienta *Labeling* (Etiquetado) crea y edita textos.



La herramienta *Wiring* (Cableado) enlaza objetos del Diagrama de Bloques y asigna a los terminales del conector del VI los controles e indicadores del panel frontal.



Object Pop-up. Desplega el menú asociado al objeto sobre el que se hace el *pop-up*.



Scroll Window. Desplaza la pantalla en la dirección que deseemos.



La herramienta *Coloring* (Color) colorea objetos y fondos.



Set/Clear Breakpoint. Función que permite poner tantos puntos de ruptura como deseemos a lo largo del diagrama de bloques.



Get Color(Capturar Color). Permite saber de manera específica que color tiene un objetos u otros elementos.



Set Color. Misma función que *Coloring*.

4.LA PALETA DE FUNCIONES EN BridgeVIEW.

Mediante la paleta de Funciones introduciremos al Diagrama de Bloques aquellas funciones necesarias para generar el código fuente de nuestra aplicación. Cada uno de los iconos que forman la paleta de Funciones se divide en subpaletas despegables donde se encuentran dichas funciones, que incluso en ocasiones pueden ser Instrumentos Virtuales.

Si la paleta no es visible, podemos seleccionar Windows>> Show Functions Palette en el menú del Diagrama de Bloques. La siguiente figura corresponde al formato de la paleta de Funciones en su nivel más alto.



Fig. 12 Paleta de Funciones de BridgeVIEW

En primer lugar podemos observar que la paleta de funciones de BridgeVIEW es bastante similar a la de LabVIEW. Es por eso que únicamente vamos a explicar que aporta de nuevo esta paleta respecto a la de LabVIEW.

Las novedades que aporta esta paleta respecto a la de LabVIEW vienen marcadas en la siguiente figura de la paleta de funciones:



Fig. 13 Paleta de Funciones con las novedades de BridgeVIEW

Al igual que LabVIEW, BridgeVIEW tiene las estructuras clásicas que utiliza cualquier tipo de programación. Estas estructuras son la *WHILE*, la *CASE* y la *FOR*. La forma de utilizarlas y programarlas mediante el lenguaje G es la misma que en otros lenguajes de programación.

Pero una de las más importantes es la *SEQUENCE* que se utiliza para organizar programas complejos, ya que permite que nuestra aplicación se realice de una forma secuencial sin que una acción no deseada se produzca antes que otra.

Su apariencia en este tipo de programación gráfica es:

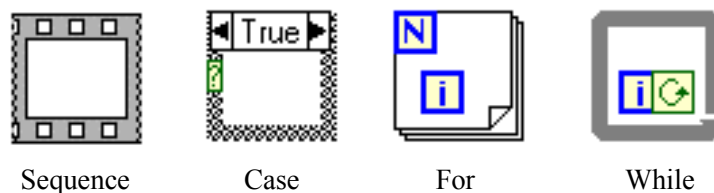


Fig. 14 Estructuras de programación en lenguaje G

Como primera novedad encontramos la subpaleta de funciones llamada **Tags**. Un *Tag* o etiqueta es una conexión con un punto de E/S del mundo real. En esta subpaleta encontramos los diferentes Instrumentos Virtuales específicos para este tipo de datos que tiene BridgeVIEW y que incluyen entre otras las operaciones de lectura y escritura de etiquetas.

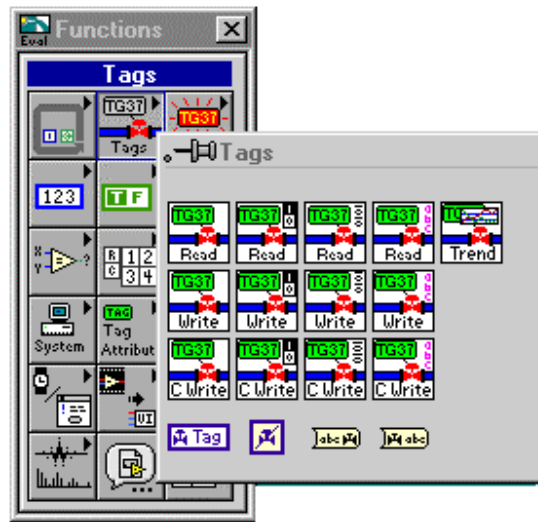


Fig. 15 Subpaleta de Funciones *Tags*

La siguiente subpaleta que encontramos es la de **Alarms & Events**. Esta incluye los diferentes Instrumentos Virtuales para controlar las alarmas de nuestra aplicación. Hay VIs para que se activan según el estados de las etiquetas y hay otros que son específicos para Indicadores Resumen de Alarmas e Indicadores de Sucesos

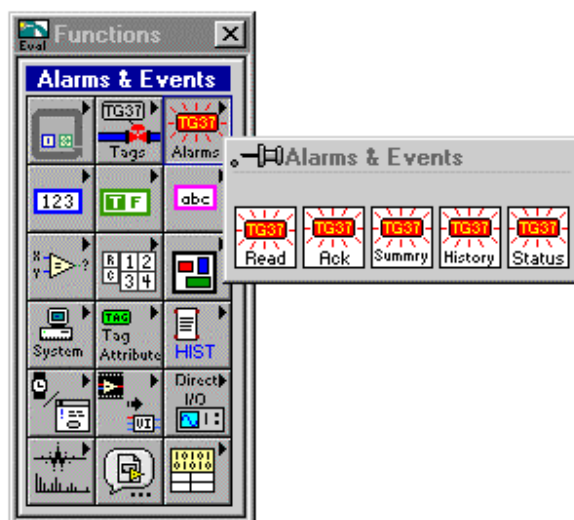


Fig. 16 Subpaleta de funciones *Alarms & Events*

La subpaleta **System** contiene un serie de VIs que repercuten directamente al sistema de BridgeVIEW. Estos VIs pueden arrancar o parar el programa, impedir el uso del mismo mediante la utilización de un login, la impresión de cualquier suceso o grafica, etc...

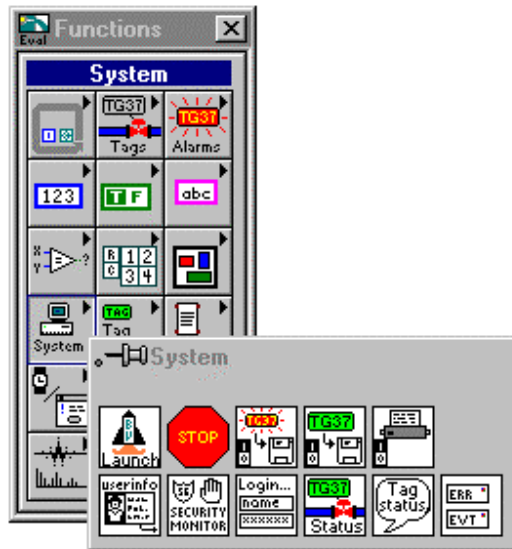


Fig. 17 Subpaleta de Funciones *System*

Otra novedad con la que nos encontramos es la subpaleta **Tag Attribute**. Aquí podemos encontrar una serie de VIs que nos sirven para conocer las propiedades de las etiquetas. De entre las funciones que podemos realizar destacamos las siguientes: obtener el valor de la etiqueta en cada momento, modificarlo, predeterminar valores de alarma...

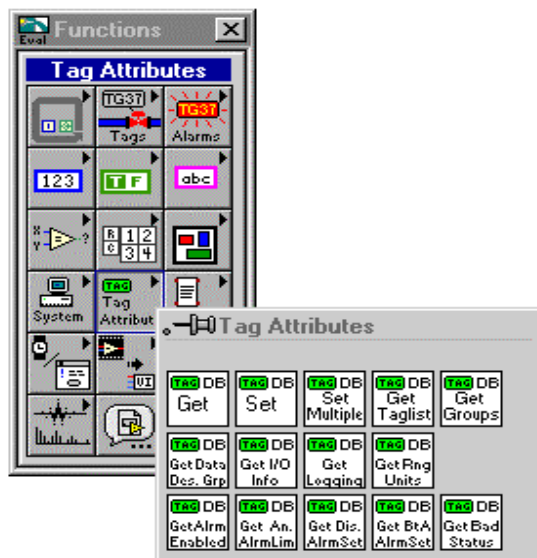


Fig. 18 Subpaleta de funciones *Tag Attributes*

En **Historical Data** nos encontramos con unas funciones que nos permiten visualizar la información sobre el sistema y las etiquetas desde que el programa ha entrado en funcionamiento. Podemos obtener la información histórica en el tiempo de valores de las etiquetas completas, desde el primer valor que ha tomado hasta el ultimo, o la podemos obtener específica, por ejemplo, el primer valor y el ultimo.

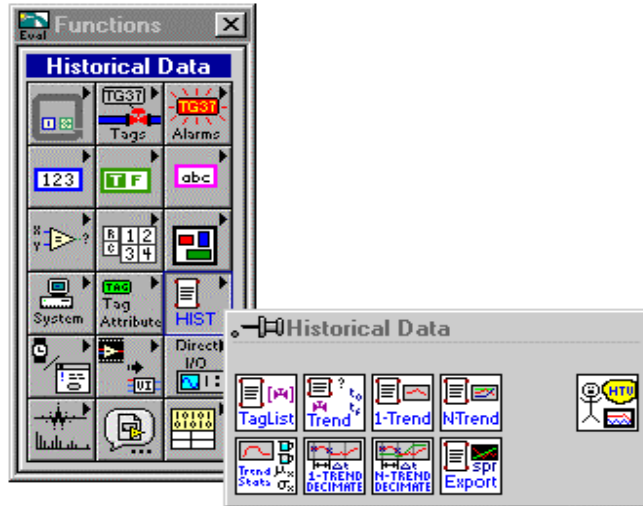


Fig. 19 Subpaleta de funciones *Historical Data*

La ultima subpaleta que diferencia la paleta de funciones de BridgeVIEW de la de LabVIEW es la de **VI Control**. En esta subpaleta tenemos un conjunto de VIs que nos sirven para trabajar con otros VIs. Entre otras funciones, podemos llamar un VI determinado en un momento del programa que así convenga, abrirlo, cerrarlo o ejecutarlo.

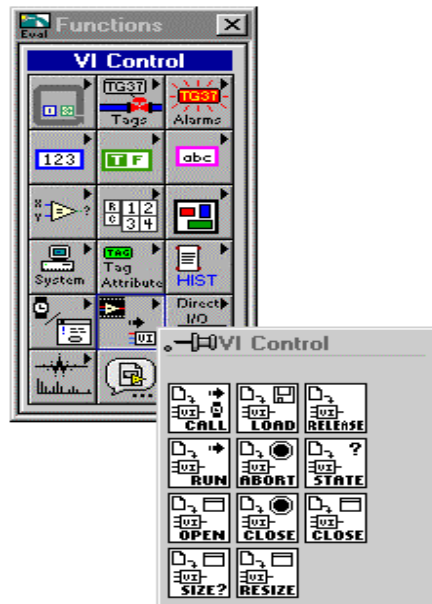


Fig. 20 Subpaleta de funciones *VI Control*

5.Aspectos avanzados de BridgeVIEW

En esta parte del trabajo veremos algunos de los aspectos específicos de BridgeVIEW que nos permitirán realizar aplicaciones de monitorización y supervisión de procesos automatizados.

Para ello haremos un resumen de algunos de los capítulos más importantes del Manual de BridgeVIEW, que se puede encontrar en la página web de National Instruments (www.natinst.com), y del que, siguiendo el orden necesario para poder realizar una aplicación, hemos hecho la siguiente clasificación de temas:

Tags (Etiquetas)

Una etiqueta o *tag* es una conexión con un punto real de entrada y salida de datos. En BridgeVIEW pueden ser de cuatro tipos:

- **Analógicas:** Una etiqueta analógica o *analog tag* es una representación del valor continuo de una conexión con un punto real de entrada y salida de datos. Este tipo de etiqueta puede variar su valor en un rango de valores de entre un máximo y un mínimo.
- **Discretas:** Una etiqueta discreta o *discrete tag* es una representación del valor de dos estados (ON/OFF) de una conexión con un punto real de entrada y salida de datos. Este tipo de etiqueta puede variar su valor entre un valor 1 (VERDADERO o *TRUE*) o un valor 0 (FALSO o *FALSE*)
- **Matriz de bits o bit array:** Una etiqueta de matriz de bits o *bit array tags* es la representación de un valor multi-bit de una conexión con un punto real de entrada y salida de datos. Este tipo de etiquetas puede abarcar hasta un máximo de 32 valores discretos
- **Cadena de caracteres o strings:** Una etiqueta de cadena de caracteres o *string tag* es la representación de un carácter ASCII conectado a un punto real de entrada y salida de datos. Este tipo de etiquetas se utiliza cuando tenemos información codificada en código ASCII, como puede ser la información leída a través de un lector de código de barras, o cuando el tipo de dato que estamos utilizando no es alguno de los anteriores.

BridgeVIEW controla la Base de Datos en Tiempo Real (RTDB) que contiene información sobre todas las etiquetas del sistema, para realizar las siguientes tareas:

- Comunicarse con dispositivos
- Escalar valores de etiquetas
- Rastrear el estado de alarmas y eventos asociados con etiquetas y errores del sistema
- Adquirir los valores de alarmas, eventos y mensajes del sistema.

Mediante el Editor de Configuración de Etiquetas o *Tag Configuration Editor* podemos definir y configurar nuestras propias etiquetas. El sistema de BridgeVIEW permite la adquisición de

datos de los valores de estas etiquetas y puede calcular automáticamente alarmas, si están configuradas correctamente.

En la siguiente figura podemos ver el aspecto de este editor de configuración:

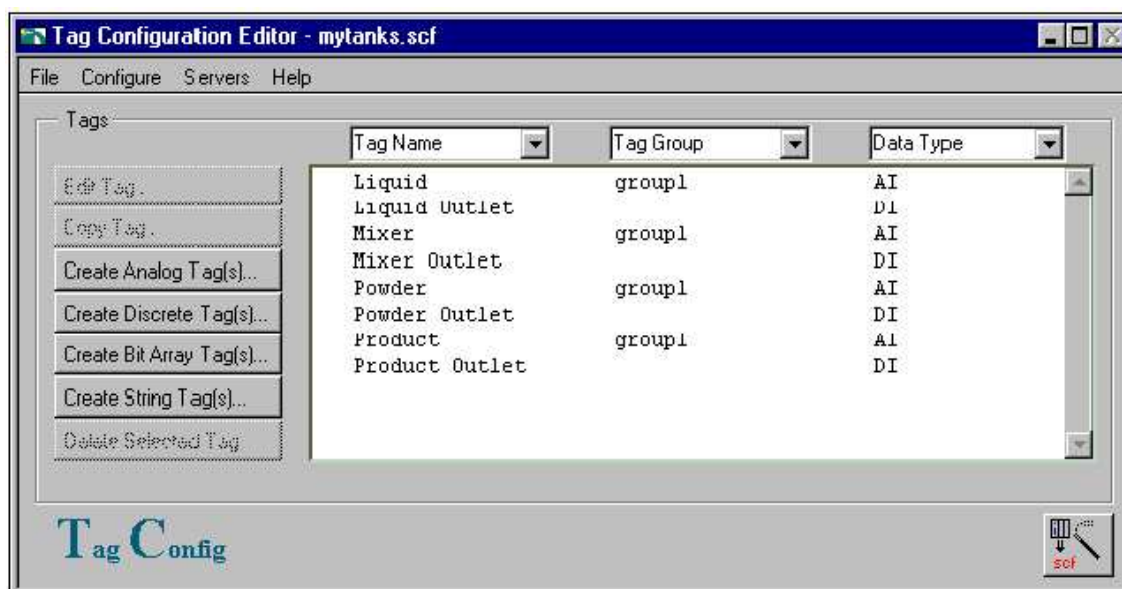


Fig. 21 Editor de Configuración de Etiquetas

El Editor de Configuración de Etiquetas o *Tag Configuration Editor* es la herramienta con la que podemos configurar todos los parámetros que controla nuestro sistema BridgeVIEW. Su misión principal es la de configurar todas las etiquetas de nuestro sistema. Recoge toda la información del sistema y la almacena en ficheros con la extensión .scf (SCADA Configuración File). Con él podemos especificar lo siguiente:

- Etiquetas usadas en el sistema
- Direcciones de los ficheros donde se almacenan los datos y sucesos históricos.

Los ficheros .scf no contienen ninguna información sobre los instrumentos virtuales porque no son específicos de una única aplicación. Por eso, múltiples aplicaciones pueden ser ejecutadas utilizando las mismas etiquetas entre ellas.

Cuando se lanza el Editor de Configuración de Etiquetas, automáticamente se abre el fichero .scf abierto por última vez, y únicamente puede haber abierto uno de estos ficheros a la vez cuando el sistema de BridgeVIEW se está ejecutando.

Human Machine Interface o Interface Hombre Máquina (HMI)

Un HMI es el interface a través del cual un operador, que es en definitiva el usuario final del sistema, interactúa con el sistema BridgeVIEW y el proceso que controla y monitoriza.

Para monitorizar los cambios producidos en etiquetas configuradas se pueden construir una o más aplicaciones HMI.

Para construir un HMI es necesario saber:

- Construir paneles frontales y diagramas de bloques
- Usar los diferentes tipos de datos de etiquetas (analógicas, discretas...)
- Usar controles e indicadores
- Usar las estructuras *Sequence* y *While Loop*
- Usar la librería de funciones *Time and Dialog*

Al construir un HMI podremos hacer lo siguiente:

- Leer y escribir valores de etiquetas
- Visión y reconocimiento de estados de alarmas y sucesos
- Visualización histórica y en tiempo real de información
- Leer la configuración de etiquetas y de información de seguridad
- Controlar el sistema BridgeVIEW
- Acceder a atributos de etiquetas, así como modificarlos
- Controlar etiquetas de salida

Para construir un HMI será necesario configurar las etiquetas que formarán el sistema, crear el panel frontal que nos servirá como *interface* y usar el *HMI G Wizard* que nos será de gran ayuda para construir el diagrama de bloques.

Con el *HMI G Wizard* se simplifica en gran medida la dificultad de construir el diagrama de bloques de nuestras aplicaciones, ya que asocia un control o un indicador del panel frontal con una etiqueta de nuestro sistema, y genera el subdiagrama que corresponda a nuestras especificaciones.

De esta manera una etiqueta de entrada analógica estará asociada a un control numérico, una etiqueta analógica de salida estará asociada a un indicador numérico, mientras que etiquetas de control discretas o alarmas de reconocimiento lo estarán a controles

booleanos. Así sucederá para todas las etiquetas y alarmas que tengamos configuradas tal y como se puede observar en la tabla 4-1 del Manual de Usuario de BridgeVIEW y que esta incluida en el Anexo.

Para llamar a un *Wizard*, lo único que tendremos que hacer será colocarnos desde el Panel Frontal encima del objeto deseado y seleccionar la opción *HMI G Wizard*. Así para una etiqueta de entrada analógica aparecerá la siguiente figura:

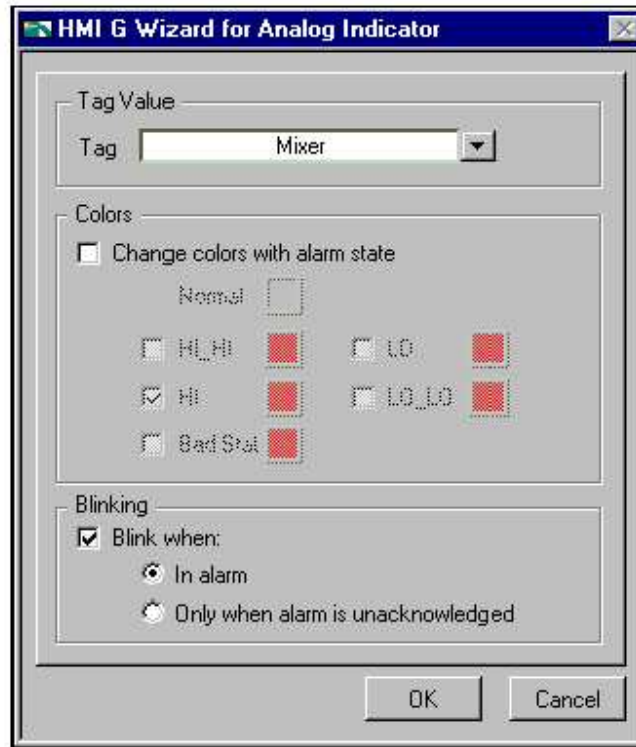


Fig. 22 HMI G Wizard

Con este *Wizard* podemos asociar un objeto del Panel Frontal con una etiqueta y ajustar varios parámetros. Cuando seleccionemos **OK** en la caja de dialogo, el *Wizard* creará automáticamente un código en el Diagrama de Bloques de acuerdo con los parámetros que hayamos seleccionado.

Por ejemplo, usando el *HMI G Wizard* con un Indicador Analógico como el de la figura anterior, aparecerá el siguiente código en el Diagrama de Bloques:

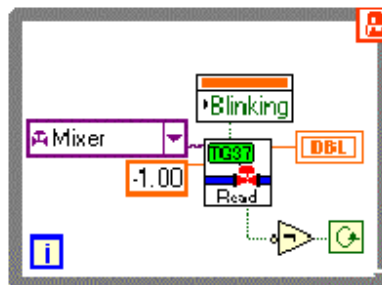


Fig. 23 Código del Diagrama de Bloques a partir del HMI G Wizard

Alarmas y Sucesos (Alarms and Events)

Una alarma es una situación anormal perteneciente a una etiqueta. En BridgeVIEW las alarmas son generadas a través de cambios en el valor o el estado de las etiquetas.

Un suceso es aquella situación que ha sucedido en el sistema de BridgeVIEW. Los sucesos pueden ser divididos en dos grupos: aquellos que están relacionados con etiquetas individuales y aquellos que pertenecen al conjunto del sistema de BridgeVIEW. Un ejemplo de suceso relacionado con etiquetas individuales incluye el cambio del estado de la alarma de una etiqueta. Algunos ejemplos de sucesos relacionados con el sistema pueden ser que se ha producido una adquisición de datos o que un usuario ha entrado en el sistema.

Para todos los tipos de datos (analógicos, discretos, matrices de bits o strings) si el dispositivo (PLC, microcontrolador o cualquier otro tipo de dispositivo) devuelve un mal estado la etiqueta pasa inmediatamente a alarma de Mal Estado o *Bad Status alarm*.

Las etiquetas analógicas si superan un valor numérico predeterminado (llamado límite de alarma o *alarm limit*) pasan a un estado de alarma.

Al igual que pasa con las interrupciones en un PC podemos decidir cuales de las alarmas de nuestro sistema tendrán mayor o menor prioridad. El rango de prioridades va desde la 1 (la más baja) hasta la 15 (la más alta). Desde nuestro HMI podremos elegir cual será la prioridad que asignaremos a cada una de las alarmas.

Un resumen de alarmas es una colección de todas las alarmas que forman nuestro sistema. Podremos informar sobre el estado de alarmas a nuestro HMI usando el Indicador de Resumen de Alarmas o *Alarm Summary Display* que se encuentra en la paleta de controles **Alarms and Events** del Panel Frontal y en la librería de funciones Resumen de Lectura de Alarmas VI o *Read Alarm Summary VI* que podremos encontrar en la paleta de funciones **Alarms and Events** del Diagrama de Bloques.

De igual manera ocurre con los sucesos, pero en este caso no se tratará de un resumen sino de una historia de sucesos o *event history*, y para informar a nuestro HMI de los sucesos se utilizará el Indicador de Historia de Sucesos o *Event History Display* de la paleta de controles del Panel Frontal **Alarms and Events** y la librería Historia de Lectura de Sucesos VI o *Read Event History VI* que se puede encontrar en la paleta de funciones del Diagrama de Bloques.

En la siguiente figura podemos ver un Indicador Resumen de Alarmas sacado de un panel frontal:

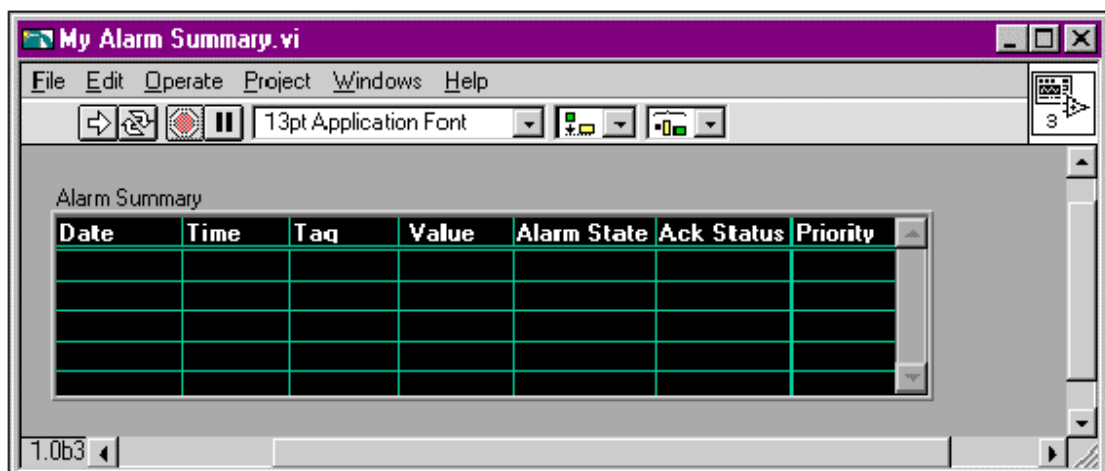


Fig. 24 Indicador Resumen de Alarmas en el Panel Frontal

En el momento que se produzca una alarma el operario que controla el sistema que monitoriza BridgeVIEW podrá dar una señal al programa de que se ha dado cuenta de que se está produciendo la situación de alarma. Para ello será necesario utilizar el botón de la paleta de controles del Panel Frontal **Boolean ACK** (que viene de la palabra inglesa acknowledgment que significa reconocimiento) y la librería de funciones *Acknowledge Alarms VI* de la paleta de funciones **Alarms and Events** del Diagrama de Bloques.

Cuando se produzca la situación de alarma si pulsamos el control de reconocimiento de las alarmas, el estado de reconocimiento o *acknowledgement status* del Indicador de Resumen de Alarmas cambiará del valor de no reconocimiento (UNACK), en cuyo caso el color del texto será rojo, al de reconocimiento (ACK), volviéndose el texto de color amarillo.

Servers (Servidores)

Un Dispositivo Servidor o *Device Server* de BridgeVIEW es una aplicación que se comunica e interactúa con dispositivos para la entrada y salida de datos como pueden ser PLCs, dispositivos de entrada y salida remotos y tarjetas de adquisición de datos. Los dispositivos servidores pasan el valor de las etiquetas a BridgeVIEW en tiempo real.

Además los Dispositivos Servidores manipulan e informan sobre posibles errores de comunicación y de dispositivos a BridgeVIEW. Hay diferentes servidores disponibles para diferentes familias de dispositivos y comunicaciones por red.

BridgeVIEW es capaz de soportar una gran cantidad de servidores incluyendo los *OPC Servers*, *DDE Servers* y los *IA Device Servers*.

OPC Servers están inscritos a la Fundación OPC y son suministrados por muchas compañías. Un *DDE Server* es cualquier servidor que soporta el *interface DDE Server* y finalmente, los *IA Device Servers* son un tipo de servidores desarrollados por la propia National Instruments.

Los Servidores de Simulación son tres tipos de servidores incluidos con BridgeVIEW usados en la mayoría de ejemplos y con los que podremos realizar nuestras aplicaciones de simulación. Estos tres tipos de servidores son: el *Tank Server*, el *SIM Server* y el *Cookie Server*. Con estos servidores se puede experimentar en configurar etiquetas y construir aplicaciones HMI.

Adquisición de datos históricos

BridgeVIEW dispone de una base de datos, donde se puede almacenar la información recogida del proceso a supervisar a través de la adquisición de datos, llamada *Citadel*.

Hay tres pasos a seguir a la hora de la adquisición de datos históricos:

- Se deben configurar las etiquetas para que tengan la opción de adquisición histórica activada (*historical logging enabled*).
- Se debe configurar un camino de archivo o path para la base de datos histórica.
- Se debe activar la adquisición histórica del sistema BridgeVIEW.

Hay un gran número de Instrumentos Virtuales que se pueden utilizar para manipular información adquirida. Con estos VIs podremos extraer información de forma que pueda ser representada en indicadores especiales para este tipo de datos como pueden ser los indicadores de *Historical Trend*, que se pueden encontrar en la paleta de controles **Graph** del Panel Frontal, y que nos dan una representación gráfica en el tiempo de la evolución de las etiquetas del proceso; o extraerla de forma que pueda ser representada en una hoja de cálculo.

En la siguiente figura podemos ver un Indicador de *Historical Trend*:

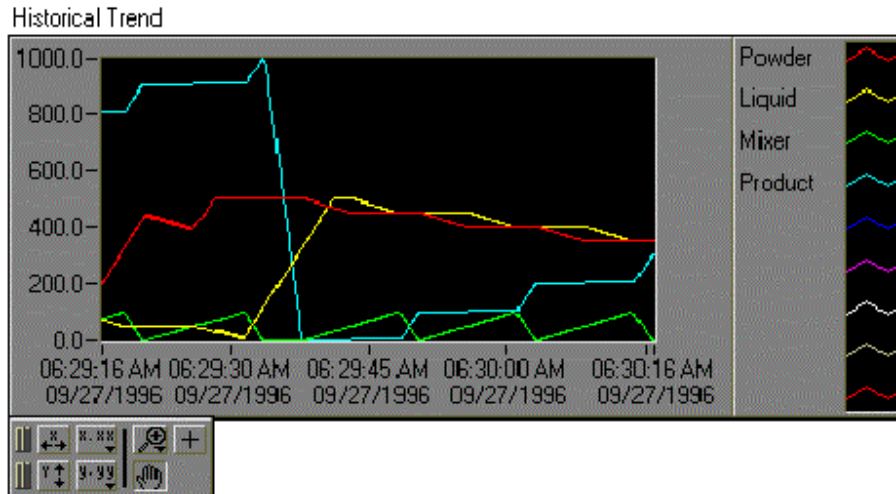


Fig. 25 Indicador Historical Trend en el Panel Frontal

Seguridad en BridgeVIEW

BridgeVIEW es capaz de crear un sistema de seguridad para controlar al personal que puede acceder a obtener información del proceso. Para ello es posible configurar que usuarios tienen acceso al sistema, que nivel de acceso y que privilegios tienen, y finalmente cual es su contraseña o *password*.

Todas estas configuraciones se realizan en el submenú **Security** del menú **Project**.

El nivel de acceso de los diferentes usuarios puede variar entre los siguientes valores: 0, 25, 50, 100, 150, 200 y 255. El usuario con un mayor número de privilegios es aquel que tiene nivel de acceso 255, mientras que el que lo tiene menor es aquel usuario de nivel de acceso 0. En la tabla 7-1 del Manual de BridgeVIEW que se puede encontrar en el Anexo, podemos ver los diferentes privilegios que podemos configurar para cada nivel de acceso, entre los que destacamos los siguientes: crear o editar etiquetas, comenzar o parar el sistema, cambiar la contraseña, añadir usuarios...

6.Creación de una aplicación con BridgeVIEW

MONITORIZACIÓN Y SUPERVISIÓN DEL NIVEL DE UN DEPOSITO

A modo de conclusión, hemos creado una aplicación a partir de los aspectos que hemos tratado anteriormente. Dicha aplicación ha consistido en la monitorización y supervisión del nivel de líquido de un depósito. Para ello hemos creado una aplicación HMI que intenta ser lo más parecida posible a como sería en la realidad. Los elementos que la forman son:

- Sistema de seguridad mediante password
- Indicador *Hopper* (tipo depósito) de la paleta de controles **Tanks**
- Indicador de alarma de la paleta de controles **Boolean**
- Indicador numérico de tiempo y fecha
- Indicador Resumen de Alarmas de la paleta de controles **Alarm and Events**
- Control **ACK** de la paleta de controles **Boolean**

El Panel Frontal de nuestra aplicación HMI es el siguiente:

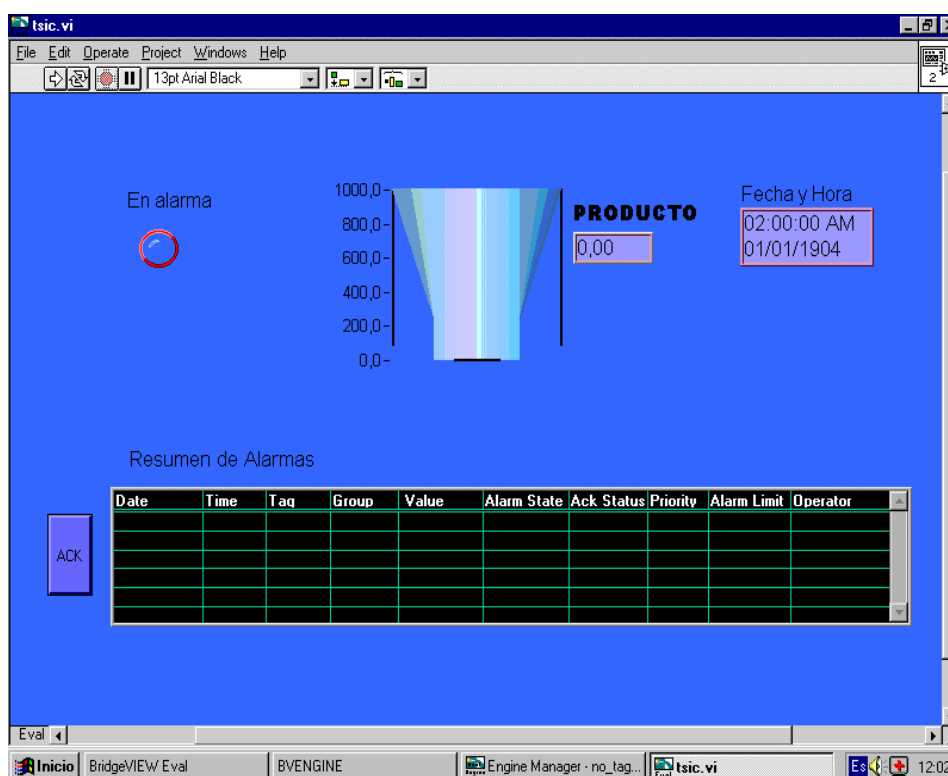


Fig. 26 Panel Frontal de nuestra aplicación HMI

Para poder ejecutar correctamente nuestra aplicación es necesario realizar las siguientes configuraciones:

- 1.- Abrir el Instrumento Virtual *Register Tanks VI* que se puede encontrar en el directorio `_server/Tanks Server`
- 2.- Ejecutar el VI
- 3.- Cerrar el VI
- 4.- Abrir el Editor de Configuración de Etiquetas seleccionando el menú **Project>>Tag>>Configuration** del Panel Frontal.
- 5.- Mediante el menú **File>>Open** seleccionar el fichero de configuración `mytanks.scf` del directorio de `BridgeVIEW Tutorial`.

NOTA: Para realizar esta configuración deberemos asegurarnos de que la versión de evaluación de BridgeVIEW disponga de este directorio que no aparece en todas las versiones ofrecidas por National Instruments

- 6.- Crear una etiqueta analógica llamada *Product* seleccionando el botón **Create Analog Tags(s)...** con las siguientes características:

Tabla de características de la etiqueta *Product*

Category	Attribute	Setting
General	Tag Name	Product
	Tag Group	group1
	Tag Description	Volume of finished product in liters
Connection	Tag Access	Input Only
	Server Name	Tanks Server
	I/O Group	ALL
	Item	tank2
Operations	Update Deadband (% of range)	1.00
	Set Initial Value	Enabled, 0.00
	Log/Print Events	Enabled
	Log Data	Enabled
	Log Deadband (% of range)	5.00
	Log Resolution (engineering units)	0.10

Category	Attribute	Setting
Scaling	Engineering Unit	Liters
	Scaling	<none>
	Raw Full Scale	1000.00
	Raw Zero Scale	0.00
	Coerce to Range	Disabled
Alarms	Enable Alarms	Enabled
	Alarm Acknowledge Mode	Auto Ack on Normal Mode
	Alarm Deadband (% of range)	1.00
	HI_HI	Enabled, Limit = 950.00, Priority = 1
	HI	Enabled, Limit = 800.00, Priority = 1
	LO	Disabled
	LO_LO	Disabled
	Bad Status Alarm	Enabled, Priority = 1

9.- Guardar la configuración seleccionando el menú **File>>Save**.

10.- Ver la configuración mediante el menú **Project>>Tag>>Browser**.

El siguiente paso a realizar será configurar el sistema de seguridad. Para ello en el menú **Project/Security/Edit User's Account** crearemos un nuevo usuario con su correspondiente password que recomendamos que sea de nivel 255, ya que así no tendrá problemas de acceso en ninguna operación al ser el nivel con mayor número de privilegios.

Una vez realizadas estas operaciones, si se han realizado correctamente, será posible ejecutar nuestra aplicación sin ningún problema.

Cuando pulsemos el botón de *RUN* lo primero que nos aparecerá será una pantalla de dialogo en la que se nos pedirá el nombre del usuario que intenta acceder al sistema y su password. Como se puede apreciar en el Diagrama de Bloques correspondiente a esta parte de la aplicación esto se ha realizado mediante el Instrumento Virtual **Invoke Login Dialog VI**, que llama a esta ventana de dialogo y no nos deja salir de ella hasta que introduzcamos el password correcto del usuario actual; y la estructura *Sequence* en su *Frame 0* para asegurarnos que lo primero que se realiza al comenzar la aplicación es que el usuario tiene permiso para manipular el sistema.

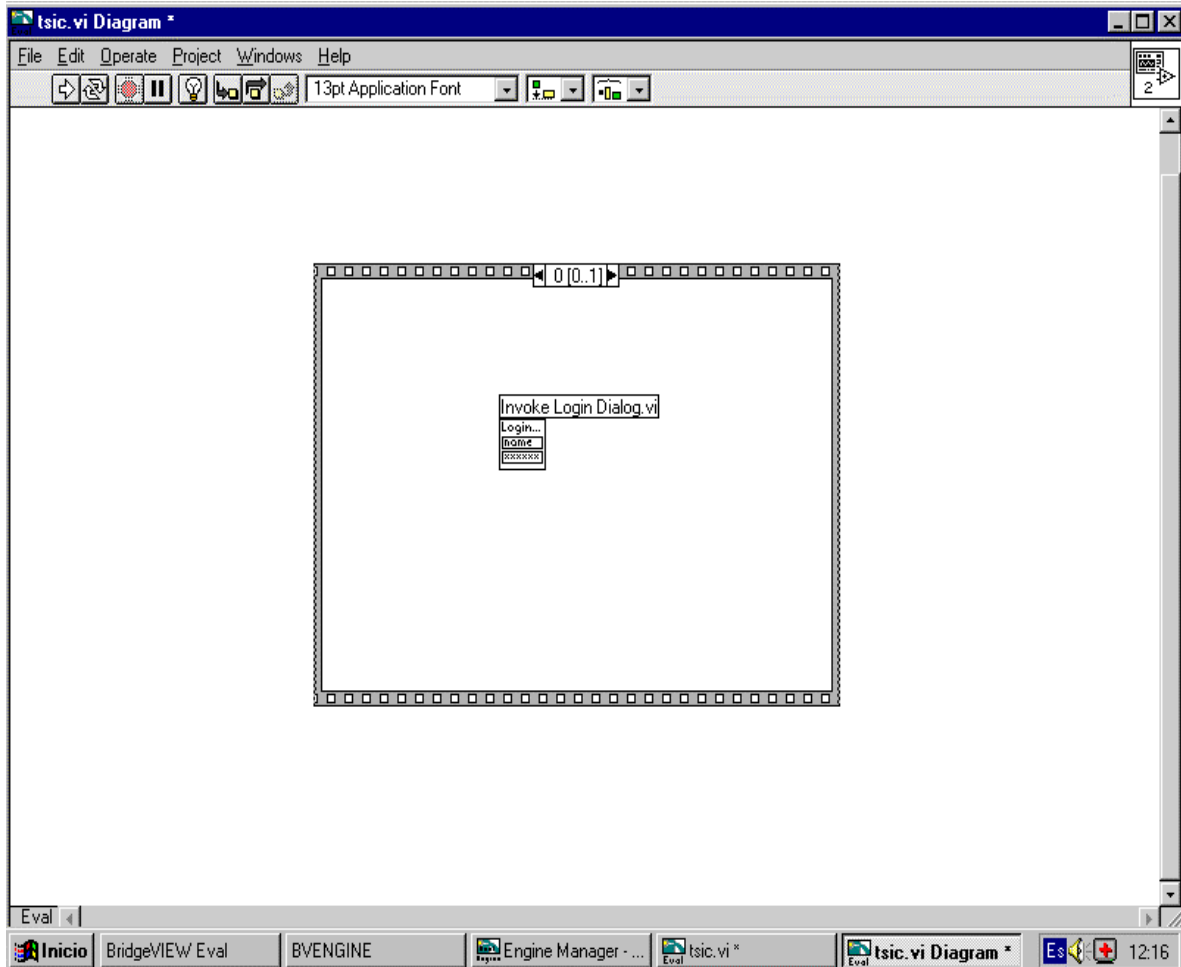


Fig. 27 Código del Diagrama de Bloques dedicado a la seguridad

Una vez hayamos introducido el password correcto, veremos como el tanque empieza a llenarse. Cuando el nivel del tanque sobrepase el valor que hemos configurado como primera alarma (800 litros) podremos apreciar que se enciende el led indicador de que estamos en estado de alarma, e instantes después podremos ver en el Indicador Resumen de Alarmas la información asociada a ella. Este será el momento de ver la función del pulsador *ACK* ya que si lo pulsamos la alarma pasará del estado de no reconocimiento o *UNACK*, que quiere decir que el usuario no esta pendiente del sistema y no se ha enterado de que se esta produciendo una situación de alarma, al estado de reconocimiento o *ACK*, con lo que le diremos al sistema de BridgeVIEW de que nos hemos enterado de que estamos en una situación de alarma.

Esta parte de la aplicación se ha realizado en el *Frame 1* de la estructura *Sequence* como puede verse en la siguiente figura; y la forma en que se ha realizado ha sido mediante el uso del *HMI G Wizard*. Lo único que hemos tenido que hacer es poner en el Panel Frontal los indicadores y controles necesarios para actuar como interface, y a partir de ahí llamar al *HMI G Wizard* (haciendo un *pop-up* sobre cada uno de ellos) que automáticamente crea en el Diagrama de Bloques el código correspondiente a la etiqueta a la que se le ha asociado el indicador o control.

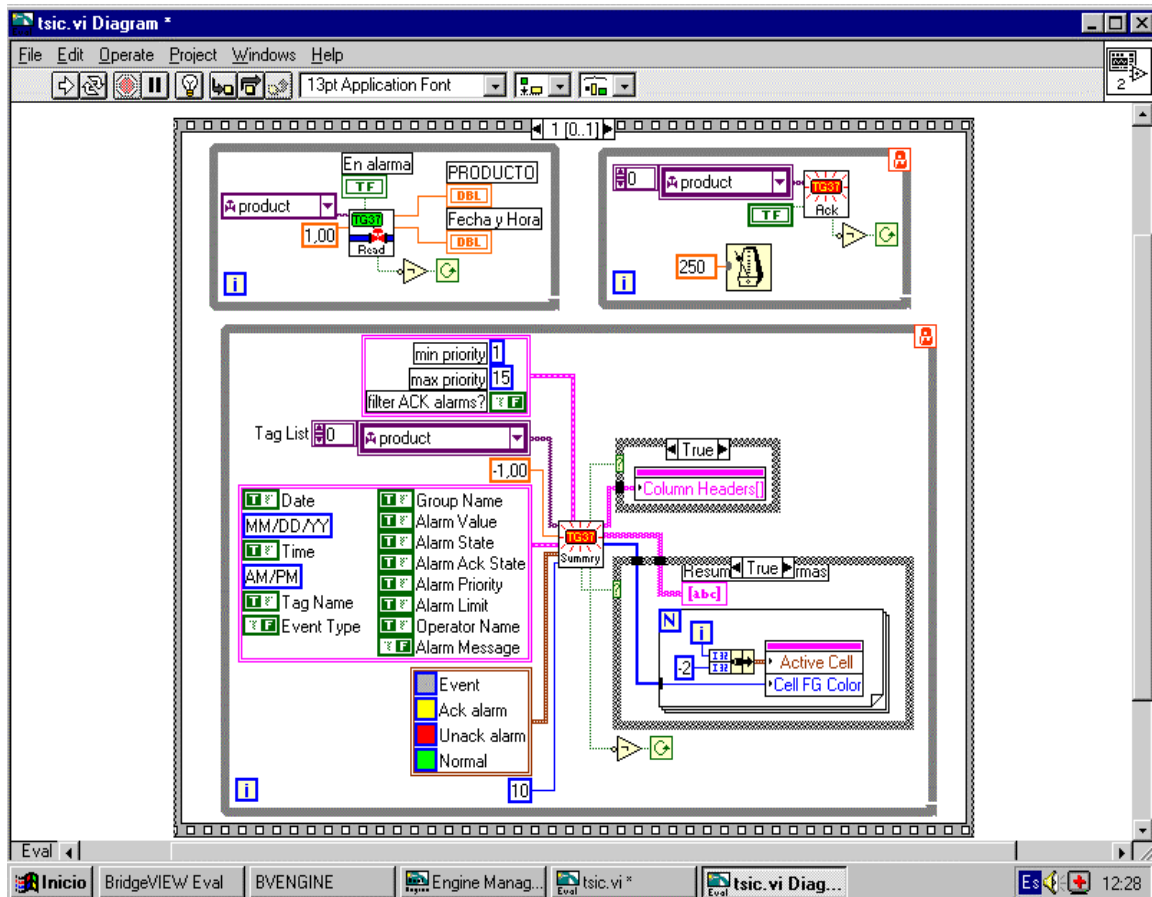


Fig. 28 Código del Diagrama de Bloques dedicado a la monitorización y supervisión

BIBLIOGRAFIA

Bibliografía básica.

- BridgeVIEW User Manual.

Archivo PDF www.natinst.com

- G Programming Reference Manual.

NATIONAL INSTRUMENTS, Edición Enero de 1998.

Bibliografía complementaria.

- Programación gráfica para el control de instrumentación

Antonio Manuel Lázaro, Editorial Paraninfo

Software utilizado.

- BridgeVIEW Evaluation 1.0