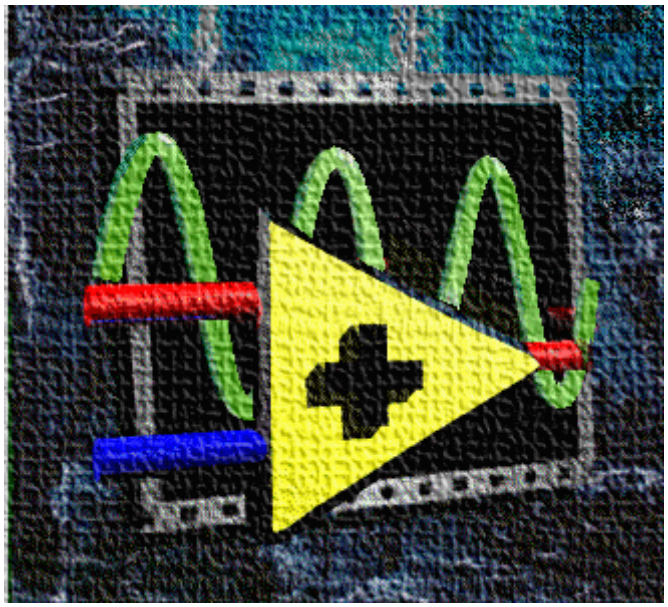
INTRODUCCIÓN A LA INSTRUMENTACIÓN VIRTUAL. PROGRAMACIÓN EN LABVIEW.



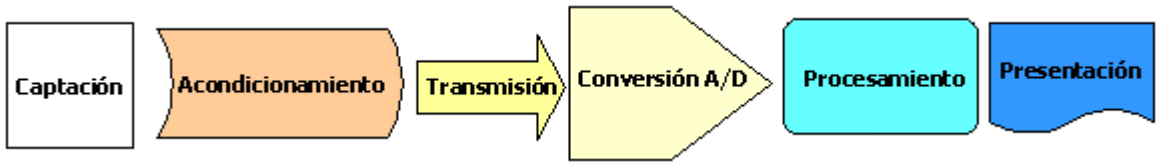
VERSIÓN 6.2

Ignacio Moreno Velasco
Pedro L. Sánchez Ortega

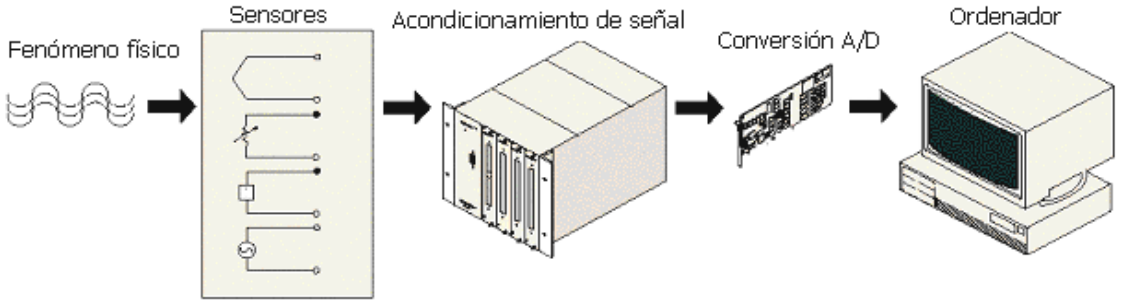
INDICE

1.- INTRODUCCIÓN A LOS SISTEMAS DE ADQUISICIÓN

El esquema general de una cadena de medida cuya misión es la adquisición de datos puede ser el siguiente:

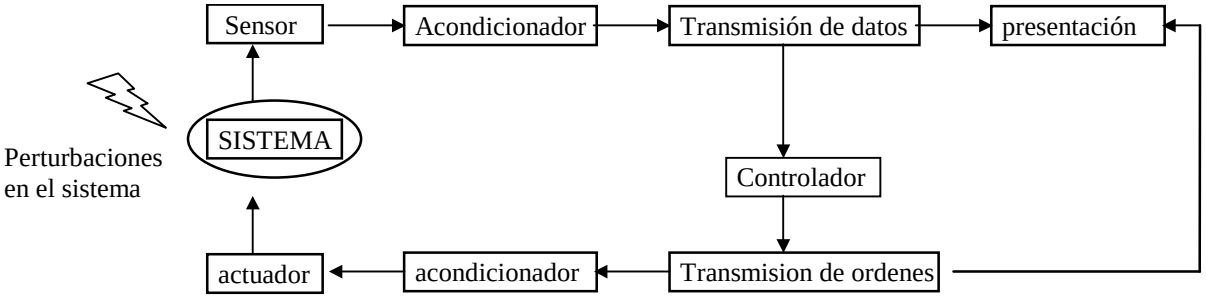


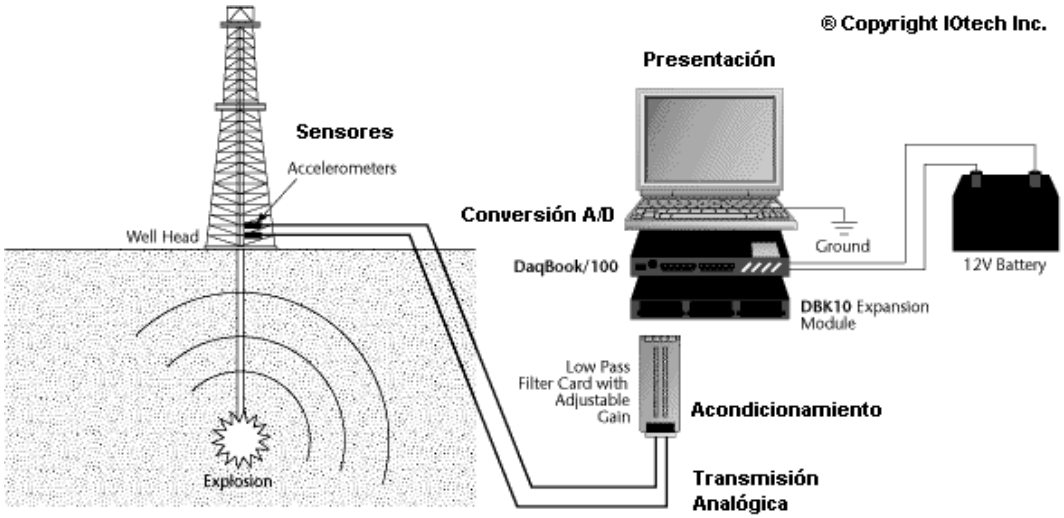
Este esquema general o alguna de sus variaciones lo encontraremos en múltiples entornos de los que cabe destacar dos: uno será el de los procesos industriales y el otro será el de ensayos y test que englobaremos bajo la denominación de entorno de laboratorio.



ENTORNO INDUSTRIAL

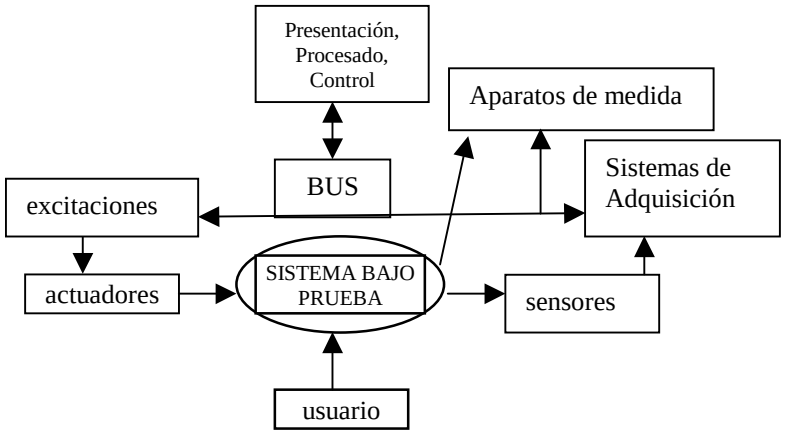
En un proceso industrial existirán varios sensores que suministran información, convenientemente acondicionada, al elemento controlador del sistema. El elemento controlador, que estará basado en algún microprocesador, recibirá la información de los sensores directamente o mediante un proceso de comunicación. Además de la presentación de dicha información en la forma deseada (generalmente gráfica) el elemento controlador dará las ordenes oportunas a los actuadores para mantener el proceso funcionando dentro de los márgnes previstos.





ENTORNO DE LABORATORIO

Para la instrumentación virtual o los procesos de laboratorio, la información puede proceder, no solo de sensores, sino también de otros instrumentos de medida (osciloscopios, multímetros, etc.) con capacidad de comunicación. Partiendo de la información recogida podemos cambiar las condiciones de la prueba, modificando parámetros de los instrumentos de medida.



En un entorno como el descrito, la tendencia actual es que sea un software especializado quien se encargue del control del sistema, coordinando el funcionamiento de los distintos elementos. Uno de estos programas software es LabView de National Instruments™. Labview permite recoger, analizar y monitorizar los datos dentro de un entorno de programación gráfico en el que se ensamblan objetos llamados instrumentos virtuales para formar el programa de aplicación con el que interactuará el usuario y que se denomina **instrumento virtual (VI)**.

Además de lo que es la propia representación de los datos en los paneles interactivos que funcionan como si se tratara de instrumentación real, permite múltiples opciones de manejo de datos, como su almacenamiento en disco y compartirlos en red o con otras aplicaciones. La interacción con otras aplicaciones se podrá realizar mediante llamadas a librerías de enlace dinámico (DLL: Dinamic Link Library) e intercambio dinámico de datos (DDE: Dynamic Data Exchange) en modo local o mediante TCP/IP en conexiones remotas. Siempre buscando independencia de la plataforma en la que hayamos realizado nuestra aplicación.

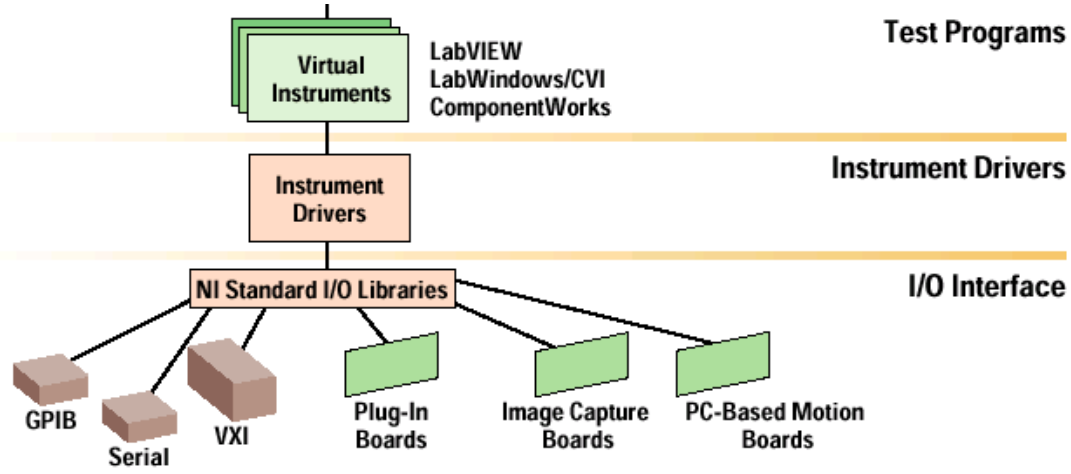
La capacidad de comunicación con otros sistemas será una cualidad importante en cualquier equipo ó sistema. Además de la comunicación mediante interfaces comunes como el RS-232 o 485, podremos utilizar otros estándares más específicos de instrumentación como el IEEE-488 más conocido como GPIB , el VXI o en entornos industriales mas específicos el CAN.

SOFTWARE DE ADQUISICIÓN DE DATOS

CONCEPTO DE INSTRUMENTO VIRTUAL

A diferencia de un instrumento real, que podemos tener en cualquier laboratorio o planta de procesos, y que queda perfectamente definido por unos mandos de control y unos elementos de representación, un instrumento virtual estará ligado al concepto de software. Este software se ejecutará en un ordenador que tendrá alojado unos elementos hardware concretos, tarjetas de adquisición de datos (analógicos y digitales), tarjetas de interfaz con los buses de instrumentación y unos canales de control también analógicos y digitales.
Nuestro instrumento virtual permitirá manejar ese hardware mediante una interfaz gráfica de usuario (IGU) que se asemejará al panel de mandos de los aparatos habituales (Osciloscopio, multímetro, etc.)
Mediante le representación en pantalla de los elementos gráficos de visualización y control que servirán de interfaz con el usuario, este observará los estados de las entradas seleccionadas en la pantalla e interactuará con las salidas directamente o mediante la ejecución de las rutinas que halla programado.

ESTRUCTURA DEL SOFTWARE



- Básicamente, el software se encargará de comunicar la interfaz de usuario del ordenador con el hardware de adquisición de datos dotando a la aplicación de la funcionalidad requerida.
- Podemos realizar una separación de las capas o partes del software: Programa de aplicación, controladores de dispositivo (drivers) y librerías de aplicación (API's).

Programa de aplicación

El programa de aplicación, también llamado instrumento virtual, consta de dos partes: interfaz de usuario y funcionalidad de la aplicación:

IGU (INTERFAZ GRÁFICA DE USUARIO)

- Permite la interacción de la aplicación con el usuario.
- Básicamente consta de controles e indicadores para visualización e introducción de datos.
- La mayoría de entornos de programación disponen de librerías de controles e indicadores creados que evitan una gran cantidad de trabajo al usuario.

FUNCIONALIDAD DE LA APLICACIÓN

- Una de las funciones básicas será la de obtener datos del hardware de forma transparente al usuario.
- La funcionalidad del programa incluye tratamiento de señal, control del flujo de programa, control de errores, etc…
- Puede implementarse en lenguajes basados en texto (Visual Basic, C++, LabWindows/CVI, etc.) o puede utilizar lenguaje gráfico como LabWiew, Snap Master, DasyLab, HP-VEE, Visual Designer de Burr Brown, etc..

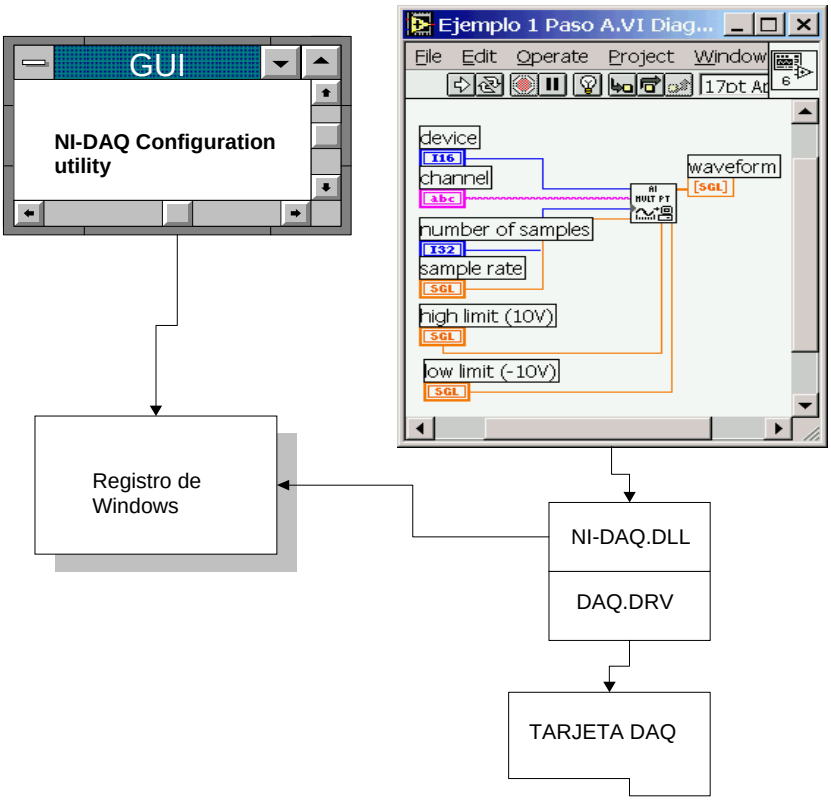
Nuestro estudio se centrará en la programación bajo el entorno de programación gráfica LabView.

PROGRAMACIÓN GRÁFICA CON LABVIEW

Los procesos programables se definirán mediante un lenguaje gráfico en vez de un lenguaje orientado a líneas de código como estamos acostumbrados normalmente.
En este tipo de programación las funciones son bloques que se interconectan entre sí, intercambiando la información.

SOFTWARE CONTROLADOR DE DISPOSITIVO

El acceso al hardware ya no se realiza mediante llamadas directas a sus registros, si no que los fabricantes proporcionan una capa intermedia que aisla al programador de detalles hardware. Esta capa intermedia facilita la comunicación entre el hardware y nuestro entorno de programación. Suele implementarse mediante DLLs, por lo que se necesita una versión específica para cada sistema operativo. (p. ej. Win16 y Win32).



Todas las tarjetas ofrecen estas librerías como complemento software. Es tan importante la documentación como la variedad y flexibilidad de las librerías.

P. EJ. NI-DAQ DE NATIONAL INSTRUMENTS

- ***nidaq32.dll***: Ocupa más de 2 MB y contiene cientos de funciones para el manejo de tarjetas de NI.
- ***nidaqcfg.dll***: Librería para la configuración de los dispositivos conectados.

2.- PRÁCTICAS DE INICIACIÓN A LABVIEW

PRÁCTICA 1: REALIZAR LA SUMA DE 2 NÚMEROS

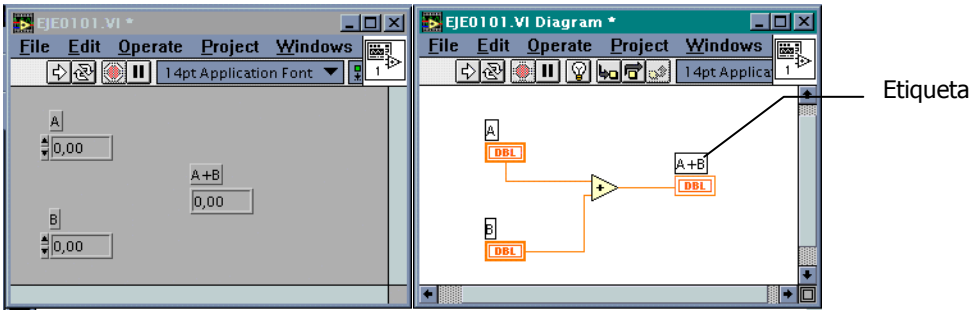
Insertar los elementos en el panel frontal.

Para ello en el panel frontal insertaremos el elemento ***Controls/Numeric/DigitalControl.*** de la paleta "Controls"
Lo haremos 3 veces, una para el elemento A, otra para el B. Para el tercero, que será la suma de ambos A+B, insertaremos un indicador en vez de un control, ***Controls/Numeric/DigitalIndicator***.

- A medida que los vayamos insertando rellenaremos la casilla de etiqueta para cada uno de ellos.
- Durante la ejecución, aumentaremos el valor de A y B mediante el dedo de la barra de herramientas
- Para mover uno de los elementos insertados podemos hacerlo mediante la flecha de la barra de herramientas, arrastrando el objeto y soltándolo en la posición deseada.

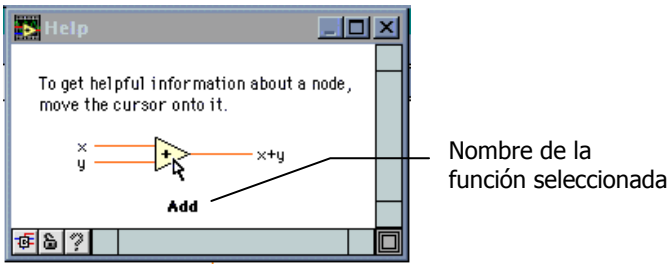
Insertar la operación de suma:

En la ventana de diagrama seleccionar ***FUNTIONS/Numeric/Add***
Distinguiremos entre control e indicador en la parte del diagrama de bloques
Las uniones que relacionan los elementos con la operación se hacen mediante el elemento ***carrete de hilo*** de la paleta de herramientas



Llamadas a la ayuda:

Mediante ***CONTROL+H*** o seleccionando la opción ***Show Help*** del menú ***HELP*** aparece una ventana de ayuda.
Situándonos sobre cualquier elemento nos informará de su utilidad y que conexiones necesita.



Ejecución del programa

Para ver el resultado de nuestro programa ejecutamos el programa pulsando con el ratón sobre el botón  .
Esto ejecutará el programa una sola vez.

- Si cambiamos los valores de los controles digitales no veremos el resultado correcto hasta que lo pulsemos de nuevo.

Si pulsamos el botón  el programa se ejecutará continuamente, por lo que si cambiamos los valores de los controles el resultado se refrescara instantáneamente.

Pulsando sobre los botones de abortar o pausa, () , respectivamente, podremos detener la ejecución definitiva o temporalmente. Para salir de la pausa volveremos a pulsar sobre ese botón.

> Pulsar sobre ***CONTROL+B*** borra de nuestro diagrama las uniones defectuosas realizadas mediante el carrete de hilo, porque no llevan a ningún sitio o porque unan elementos no relacionables.

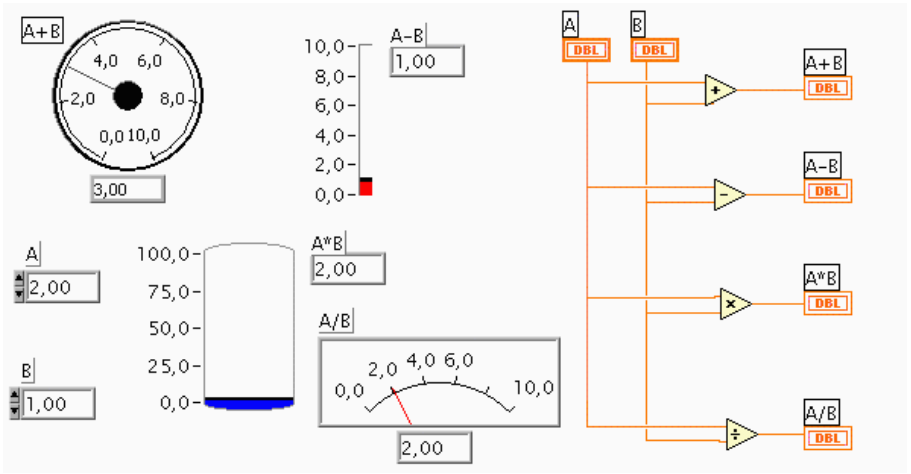
PRÁCTICA 2: OTRAS OPERACIONES Y CONTROLES

Sustitución de controles e indicadores

Sustituiremos o reemplazaremos los controles existentes por otros diferentes, como por ejemplo por ***CONTROLS/NUMERIC/Horizontal Pointer Slide***.

> Obsérvese que solamente cambiamos la parte correspondiente a la interfaz de usuario, no a la funcionalidad.

Partiendo de lo aprendido en la práctica 1, realizar las operaciones de suma, resta, multiplicación y división de las entradas A y B utilizando como salidas para los resultados distintos visualizadores, tanque, agujas, etc.

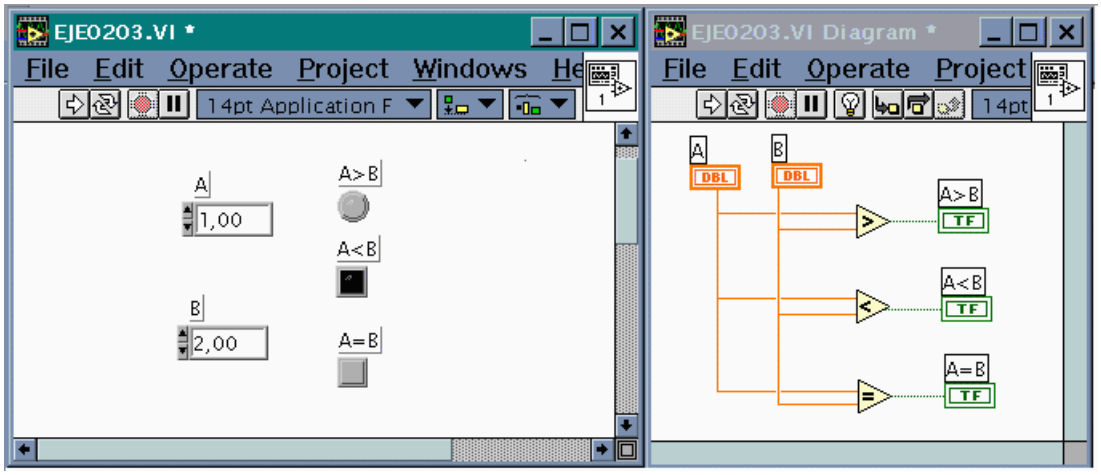


> Para cambiar la escala del depósito al valor 100, basta con sobrescribir el valor máximo con la herramienta de escritura

> Podemos realizar el cambio entre las distintas herramientas, 'flecha', 'mano', 'carrete de hilo' con la pulsación del tabulador y el espaciador.

Introducir funciones de comparación:

Las funciones de comparación se encuentran en ***FUNTIONS/COMPARISON/GREATER.***
Introducir Leds como resultado de las comparaciones, ***CONTROLS/BOOLEAN.***



Cambio de colores: podemos modificar las propiedades de color de la mayoría de los elementos del panel de control con la herramienta ***pincel***

> La alineación de las partes de un diagrama y del panel de control se realiza mediante las listas desplegables  de la barra de herramientas.

Tipos de datos:

Existen 12 representaciones para los controles o indicadores digitales:

- Precisión simple: 32 bits (SGL).
- Precisión doble: 64 bits (DBL).
- Precisión extendida (EXT): números de coma flotante.
- Número entero con signo (I8) de tipo byte (8 bits).
- Número entero sin signo (U8) de tipo byte (8 bits).
- Número entero con signo (I16) de tipo palabra (16 bits).
- Número entero sin signo (U16) de tipo palabra (16 bits).
- Número entero con signo (I32) de tipo entero extendido (32 bits).
- Número entero sin signo (U32) de tipo entero extendido (32 bits).
- Complejos de precisión simple (CSG).
- Complejos de precisión doble (CDB).
- Complejos de precisión extendida (CXT): números complejos de coma flotante.

Los límites, máximo y mínimo, dependen del tipo; un entero con signo (8 bits) estará entre -128 y 127.

Precisión:	Simple	Doble	Extendida
Número positivo máximo	3,4 E38	1,7 E308	1,1 E4932
Número positivo mínimo	1,5 E-45	5,0 E-324	1,9 E-4951
Número negativo mínimo	-1,5 E-45	-5,0 E-324	-1,9 E-4951
Número negativo máximo	-3,4 E38	-1,7 E308	-1,1 E4932

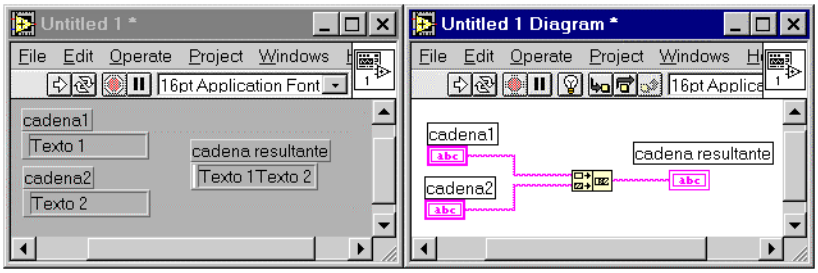
> Para realizar el cambio de la precisión de un dato invocaremos el menú contextual pulsando el botón derecho cuando estemos sobre él y seleccionando el submenú de ***REPRESENTATION*** del menú emergente.

Concatenación de cadenas alfanuméricas frente a suma de números.

La función equivalente a la suma de números es la concatenación de caracteres, que da por resultado una única cadena formada por otras simples. Esta operación es habitual cuando se trabaja con mensajes como los que circulan por el bus GPIB

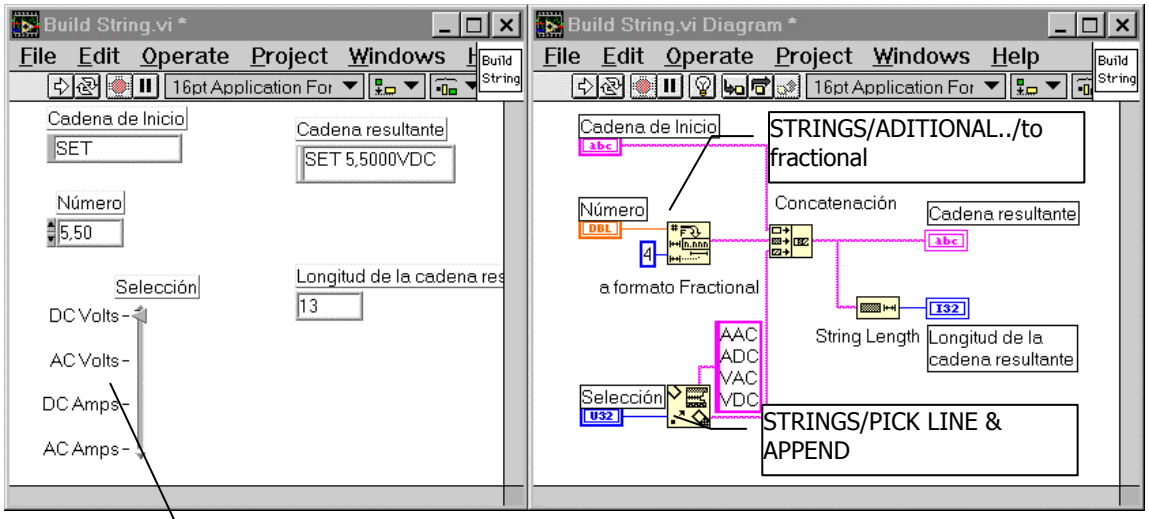
PASO A

La concatenación básica corresponde con este ejemplo, donde usamos la función ***Funcions/strings/concatenate strings***



PASO B

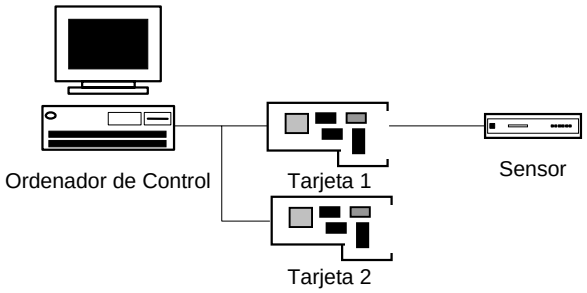
En el ejemplo siguiente concatenamos varios elementos; algunos de los cuales son el resultado de la conversión de números a cadenas de caracteres.



Para realizar un selector como el de la imagen debemos usar la propiedad ***TEXT LABEL*** en el menú contextual del selector. Una vez hecho esto, podremos añadir nuevos textos al selector, pulsando en el menú contextual que aparece en su display sobre las opciones ***ADD ITEM BEFORE*** ó ***ADD ITEM AFTER***.

PRÁCTICA 3ª: ADQUISICIÓN Y PROCESAMIENTO BÁSICOS

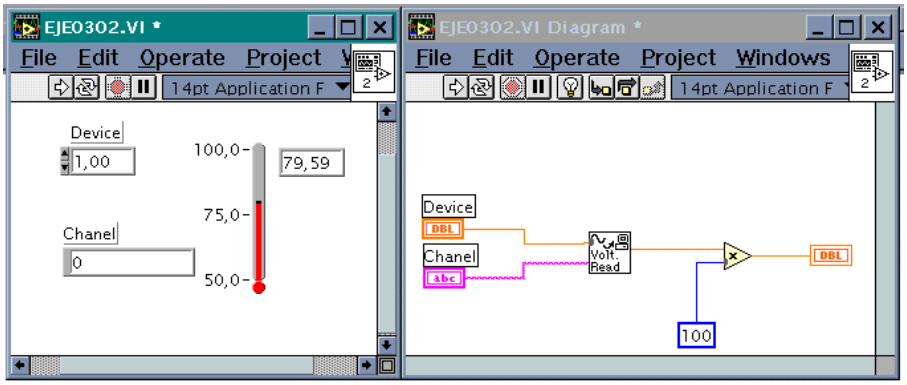
La adquisición aunque simulada correspondería con un esquema hardware donde un sensor está conectado a uno de los canales de una de las tarjetas de adquisición de datos disponibles en el ordenador. El esquema sería el siguiente:



Insertar la función : ***FUNCTIONS/TUTORIAL/Demo voltaje Read.VI***

Este VI solo funciona si seleccionamos los valores ***0*** ó ***1*** en el control ***Device*** y el control ***Chanel*** dando error en cualquier otro caso. (i.e. Device 0 = Tarjeta 1, Device 1 = Tarjeta 2).

TÉCNICAS DE DEPURACIÓN:

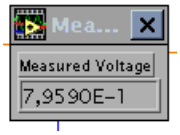


Propuesto: ***Dibujar la curva de calibración del sensor.***

Seleccionando el botón  de la ventana de diagrama se ejecutará la aplicación en modo depuración.

Podemos insertar una ***punta de prueba*** en cualquier cable del diagrama para visualizar el valor en dicho punto.

- Para crear esta punta de prueba, seleccionamos la herramienta  del menú de herramientas, pulsando con el ratón en aquellos puntos de cable donde deseemos saber el valor de la variable.

- En el momento de ejecución nos aparecerá en el diagrama una ventana con el valor. 

Recordemos que para obtener ayuda sobre una determinada función podemos activar la ayuda. Se la llama mediante CONTROL-H. y al situarnos sobre un elemento nos da la información referida a el, conexiones y tipos de datos que utiliza.

CREACIÓN DE SUBVI'S

Son el equivalente a las subrutinas en los lenguajes de programación basados en texto.Con ellos conseguimos una programación mejor estructurada y por tanto más legible así como evitar la repetición de código. La diferencia respecto a las tradicionales subrutinas es que un subVI puede ejecutarse de forma autónoma sin necesidad de que esté incluido en un VI.

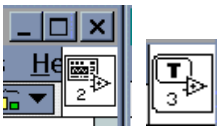
Al hacer doble click sobre un subVI se abre el panel de control correspondiente a dicho subprograma.

Para salvar un VI como fichero lo haremos mediante las distintas opciones del menú ***FILE***, ***SAVE*** y ***SAVE AS***. Deberemos poner nosotros la extensión ***.VI*** ya que por defecto el programa no añade extensión alguna.

Crear un subVI

Partiendo de un VI como el del ultimo ejemplo, crearemos un subVI para utilizarlo posteriormente en otras partes de nuestros VI's. Tendremos que definir unas entradas y unas salidas para posteriormente efectuar las conexiones en el diagrama.

- Haciendo doble click (doble pulsación) sobre el icono de la parte superior derecha del panel de control podemos editar el dibujo que identificará a nuestro subVI.



- Si una vez modificado y dejado como definitivo, pulsamos con el botón derecho y elegimos la opción ***Show conector*** veremos la disposición de conexiones de nuestro subVI:

- Pulsar el botón derecho para ver los distintos tipos de ***patterns*** (plantillas de conexión) y elegir el que se adapte a nuestras entradas y salidas; normalmente las entradas estarán a la izquierda y las salidas a la derecha.
- En la ventana panel de control seleccionar la posición dentro del ***pattern*** y con el carrete de hilo el control correspondiente de entrada o salida. (Si esta seleccionado como entrada o salida habrá cambiado del blanco al gris en el pattern elegido por nosotros).
- Una vez realizadas las conexiones y modificaciones del icono podemos guardarlas.

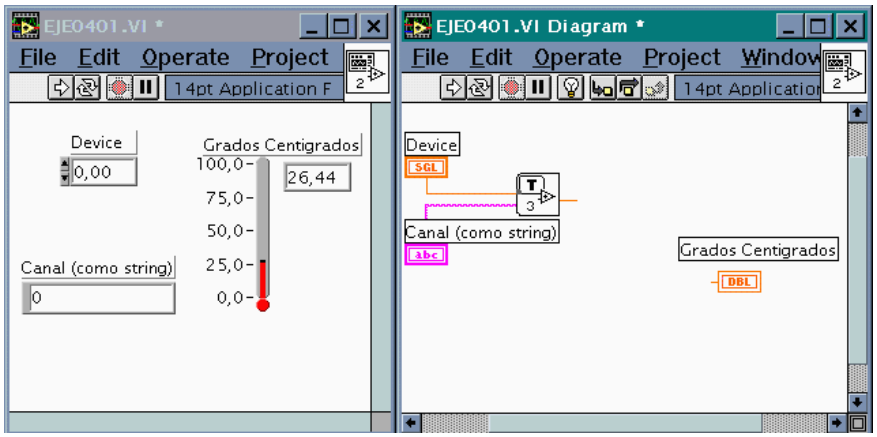
PRÁCTICA 4ª: UTILIZACIÓN DE SUBVI'S

Partiendo del subVI que acabamos de crear y que muestra la temperatura en grados Fahrenheit, realizar un VI que visualice la temperatura en grados centígrados.

Introducir el subVI creado en la práctica 3ª

Lo haremos mediante el botón de la paleta de funciones '***Select a VI...***'. Lo seleccionaremos de la lista de ficheros y lo pegaremos en la ventana ***diagrama*** de nuestro instrumento virtual.

Completar el diagrama de bloques teniendo en cuenta que: ***Tª (ºC) = [Tª (ºF) - 32] * 5/9***



OPCIONAL:
Añadir un indicador de grados Kelvin
ºK = ºC + 273

OPCIONAL:
Crear un subVI con tres salidas: ºC, ºF y K.

En vez de crear manualmente un control para cada entrada y un indicador para cada salida, es posible realizar esta tarea automáticamente:

- Mediante el carrete de hilo, pulsar con el botón derecho sobre la entrada o salida y elegir la opción ***create control*** o ***create indicator*** .
- Según el caso el programa crea automáticamente el tipo necesitado incluso con su etiqueta.

Si no hemos puesto la etiqueta a algún elemento deberemos seleccionar el elemento con el botón derecho y elegir la opción ***show label***, una vez en pantalla introducir el valor deseado.

Siguiendo la ejecución del programa en modo de depuración  podemos ver como las funciones esperan de izquierda a derecha a que se vayan generando los datos que necesitan para completar la operación.

Un SubVI puede llamar a múltiples VI's dando lugar a una jerarquía que podemos visualizar gráficamente. Para ello existe la opción ***show VI hierarchy*** del menú ***Project***.

PRÁCTICA 5ª: SECUENCIA WHILE-LOOP

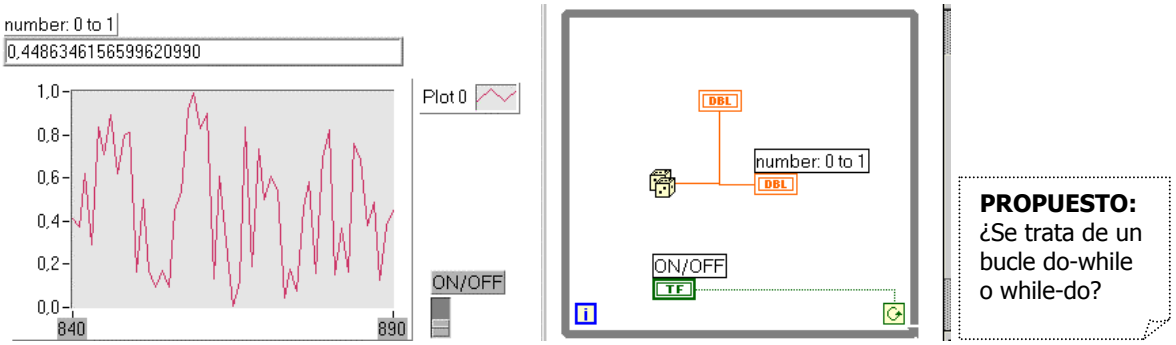
La secuencia ***While-loop*** permite repetir las acciones que situemos dentro del bucle hasta que deje de cumplirse una condición que nosotros establezcamos.

PASO A: GENERACIÓN DE NÚMEROS ALEATORIOS

Como ejemplo generaremos un número aleatorio entre 0 y 1 mediante la función ***FUNTIONS/NUMERIC/Ramdom Number***. El resultado lo representaremos mediante una gráfica en el panel de control: ***CONTROL/GRAPH/Wave form Chart***. La ejecución debe parar cuando el usuario pulse un botón ***ON/OFF.***

1º Dibujar el bucle *while:*

Para ello, situaremos la función dentro de un bucle While que se ejecutará como mínimo 1 vez hasta que la condición procedente del botón sea ***FALSE***.



PROPUESTO: ¿Se trata de un bucle do-while o while-do?

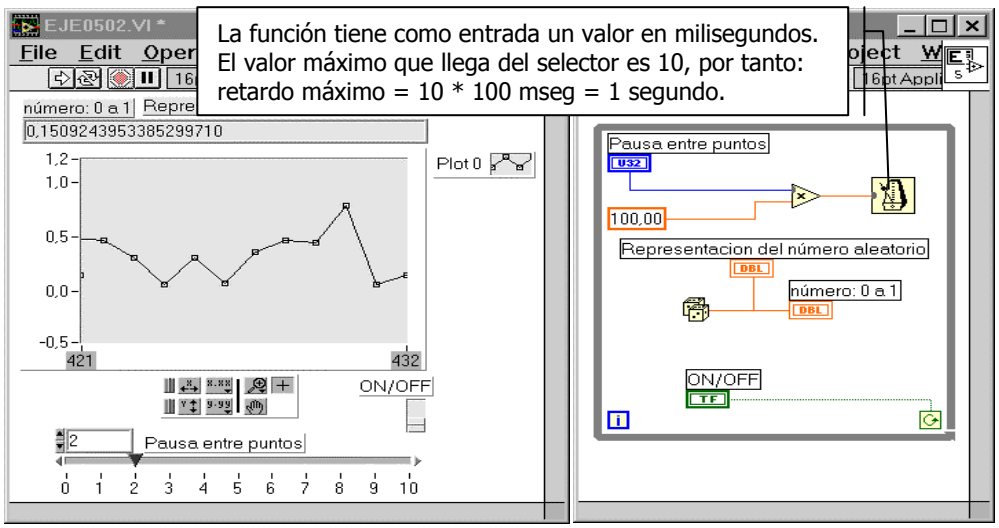
2º Modificar el número de decimales

El nº de decimales del visualizador se modifica con la propiedad ***FORMAT & PRECISSION*** de su menú contextual.

> **Importante**: reseñar la diferencia entre parar un programa correctamente, una vez acabadas las acciones contenidas en un bucle o abortar la ejecución mediante el botón sin ningún control sobre las acciones que han sido ejecutadas y las que han quedado pendientes.

PASO B: AÑADIR RETARDO ENTRE ITERACIONES

Podemos incluir un retardo entre la generación de uno y otro punto mediante la función ***FUNTIONS/Time & Dialog/Wait Until Next ms Multiple***, tal y como puede verse en la siguiente figura:

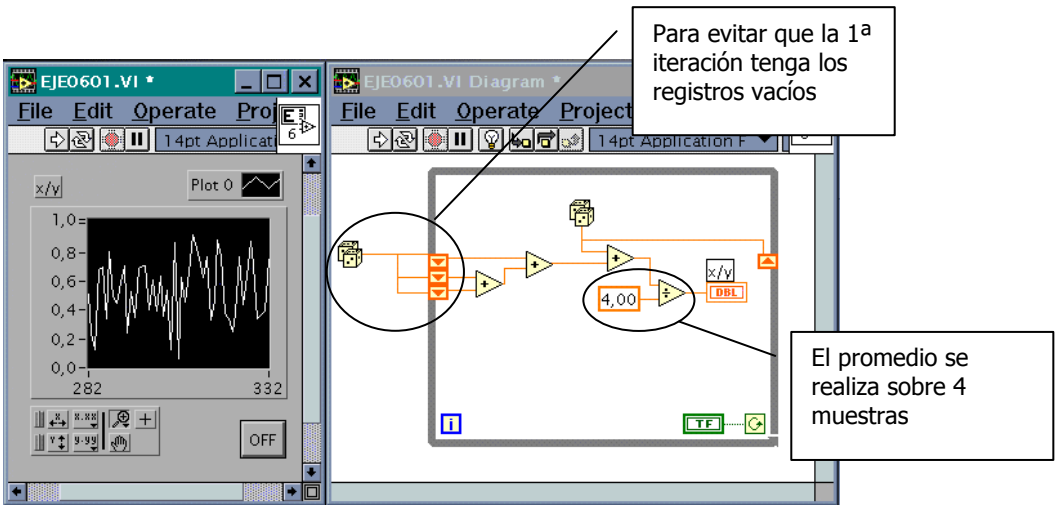


La función tiene como entrada un valor en milisegundos. El valor máximo que llega del selector es 10, por tanto: retardo máximo = 10 * 100 mseg = 1 segundo.

PROPUESTO: ¿Qué se ejecuta antes, la pausa o el número aleatorio?

PRÁCTICA 6ª: REGISTROS DE DESPLAZAMIENTO

Veremos la aplicación de los registros de desplazamiento (***Shit registers***) en el cálculo de promedios.



PASO A: AÑADIR LOS REGISTROS DE DESPLAZAMIENTO:

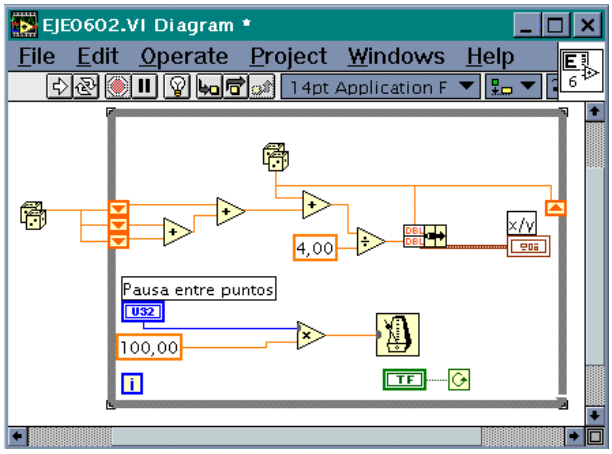
- En la zona izda. del bucle pinchar con el ratón y pulsar el botón derecho, seleccionar ***Add Shif Register***.
- Después en la zona derecha pinchar y seleccionar ***Add Element***

Para comprender mejor el efecto del registro de desplazamiento:

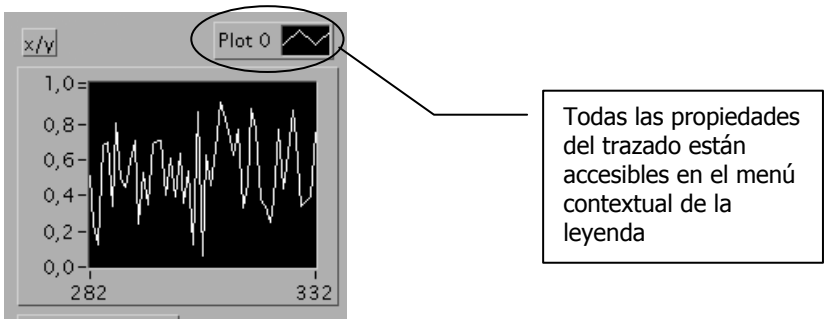
- Activar el modo de depuración
- Observar como evolucionan los valores.

PASO B: VISUALIZAR DOS FUNCIONES EN EL MISMO GRÁFICO

- Para visualizar simultaneamente el gráfico corresponiente a la media y al valor original; lo haremos mediante la opción ***FUNCTIONS/CLUSTER/BUNDLE.***

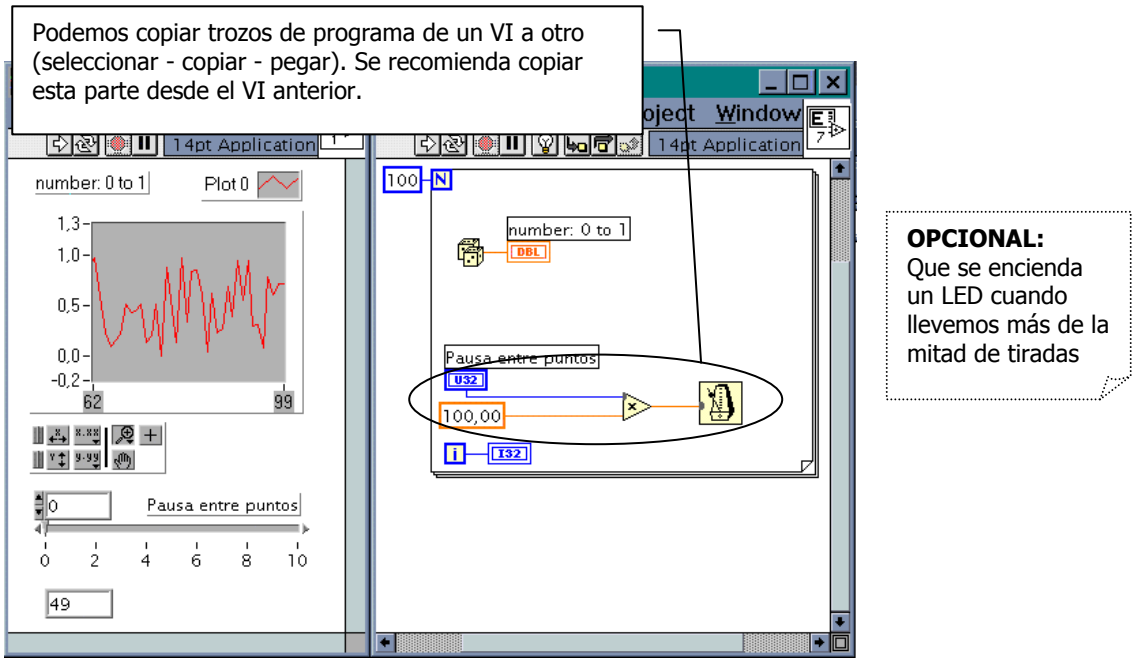


Cambiar el color, grosor y tipo de trazo de los gráficos:



PRÁCTICA 7ª: BUCLE FOR.

Un bucle For (***For-loop***) permite la ejecución del código que situemos dentro del bucle un número de veces predeterminada.



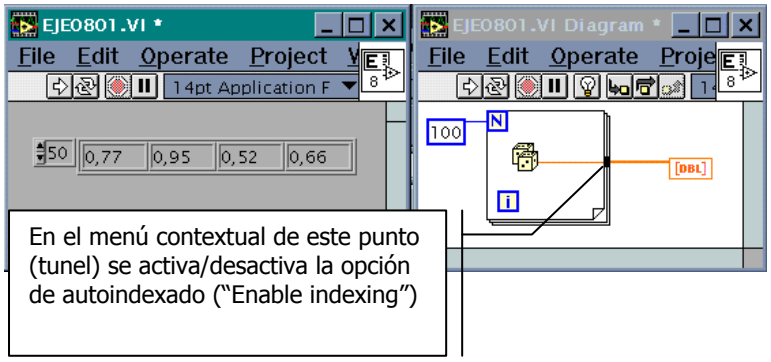
PRÁCTICA 8ª: MATRICES.

PASO A: GENERACIÓN

Construiremos una matriz que contendrá 100 números aleatorios generados por el VI en forma de dados. Para ello, en el panel de control:

- Primero, insertar un visualizador de tipo ***array***: ***CONTROL/ARRAY & CLUSTER/Array***.
- Después debe introducirse un visualizador numérico dentro del control ***array*** (soltándolo encima).

Los elementos del array se almacenarán en el túnel (punto negro en el marco de la estructura for-loop) si tenemos habilitada la opción de autoindexado, sino sólo el último dato será pasado al exterior del bucle.

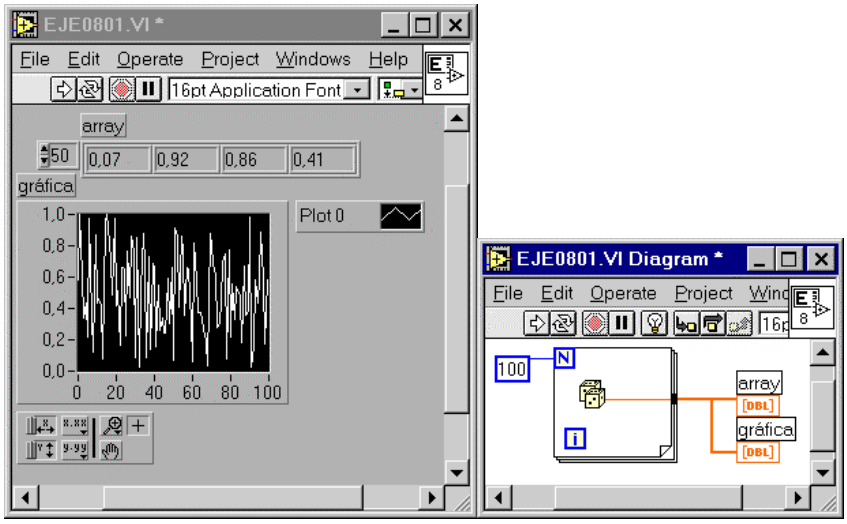


En el menú contextual de este punto (tunel) se activa/desactiva la opción de autoindexado ("Enable indexing")

OPCIONAL: Realizar el promedio de 10 números aleatorios mediante un registro de desplazamiento. **Ojo,** debe inicializarse el registro a cero.

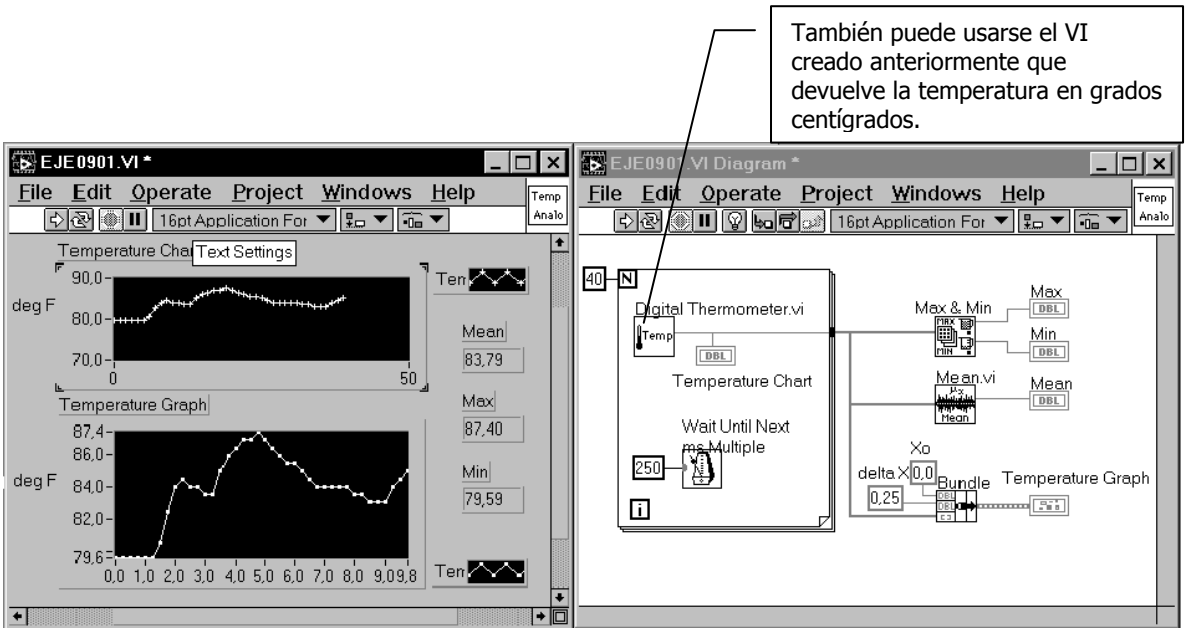
PASO B: VISUALIZACIÓN GRÁFICA

Añadir un gráfico ***CONTROLS/GRAPH/Wave form Graph*** para la visualización gráfica del array. Quedará como se muestra en el panel y diagrama siguientes:



PRÁCTICA 9: TIPOS DE GRÁFICOS

Comparar la diferencia entre los gráficos ***CHART*** y los ***GRAPH***, calculando la media, máximo y mínimo de las temperarturas generadas aleatoriamente por el subVI denominado ***Digital Thermometer.vi*** de la librería ***Tutorial***.



- Las funciones para encontrar el máximo y el mínimo del array se encuentran en ***FUNTIONS/ARRAY/Array Max & min***
- Para calcular la media lo haremos mediante la función ***FUNTIONS/ANALISIS/Probability & Statistics/Mean***
- Para hacer el ***BUNDLE*** mas grande de forma que permita conectar más elementos, situamos el puntero en alguna de las esquinas y estiramos hacia abajo.

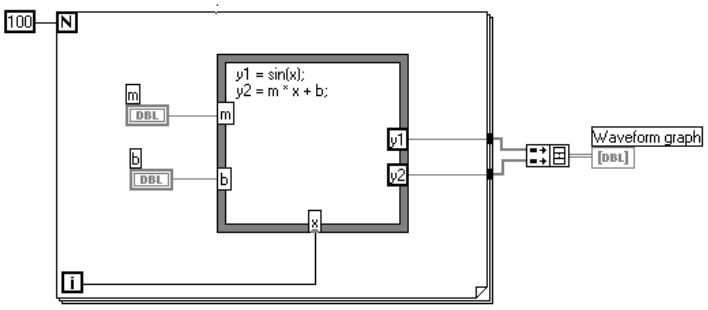
Observar la diferencia en el grosor del cable del array que sale del bucle FOR.

Tener en cuenta que incluimos 2 tipos de gráficos, *Graph* y *Chart*. Para dar sentido temporal al eje x del gráfico se debe realizar un bundle con tres elementos: la matriz de puntos, su separación en el tiempo y el origen. Como hemos fijado pausas de 250 ms entre muestras entonces delta X es 0'25 (segundos). Xo establece el origen del eje x, en este caso 0.

PRÁCTICA 10: NUDO FÓRMULA.

Permite incorporar a un VI fórmulas matemáticas definidas por el usuario.

- Lo insertamos con ***FUNTIONS/STRUCTURES/FORMULA NODO***
- Para añadir entradas o salidas lo haremos pulsando con el botón derecho del ratón en el marco del nodo fórmula y seleccionando ***add input*** o ***add output.***.
- Las fórmulas (todas las necesarias para calcular las salidas) deben terminar en punto y coma (***;***).
- Los resultados de cada fórmula dan lugar a una matriz unidimensional. La unión de esas dos matrices da lugar a una matriz de 2 dimensiones. Esa conversion se realiza mediante la función ***FUNTIONS/ARRAY/BUILD ARRAY***



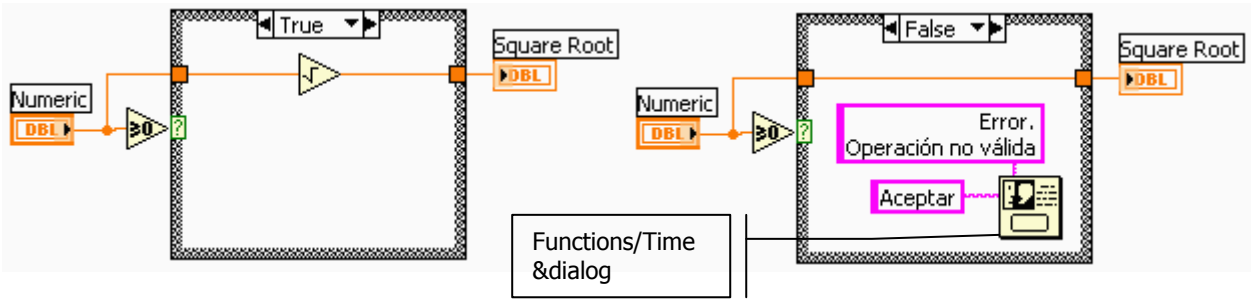
OPCIONAL:
Hallar el valor RMS de la función sinusoidal mediante el VI ***Functions/analysis /probability and statistics/RMS***

🛠 Modificar el programa de forma que el usuario pueda variar la amplitud de la sinusoide.

Un ejemplo mas completo y vistoso del uso de la estructura ***formula-node*** puede verse en la zona de ejemplos, ***Examples/General/Graphs/gengraph.llb*** en el instrumento virtual llamado ***bouncing cube.vi***

PRÁCTICA 11: ESTRUCTURA CASE

Esta estructura de programación permite que se ejecute una porción de código u otra según se cumpla o no una condición impuesta por el usuario.



OPCIÓN MÚLTIPLE DE LA ESTRUCTURA CASE.

Aunque en el ejemplo sólo existen el caso verdadero y falso, pueden ponerse tantos casos como se necesiten:

- Para añadir un caso más, ya sea antes o después deberemos seleccionar pulsando con el boton derecho del ratón en el borde ***Add case before*** ó ***Add case After*** respectivamente.
- Evidentemente la entrada al selector ya no podrá ser binaria, sino que deberá corresponder a un número entero.

3.- ADQUISICIÓN DE DATOS.

CONCEPTOS BÁSICOS DE ADQUISICIÓN DE DATOS (A/D)

VELOCIDAD DE MUESTREO

Teorema de Nyquist

Establece que para poder reconstruir correctamente una señal muestreada, la velocidad de muestreo ***fs*** debe ser al menos el doble de la mayor de las componentes de la señal muestreada:

$$fs \geq 2 * fmax$$

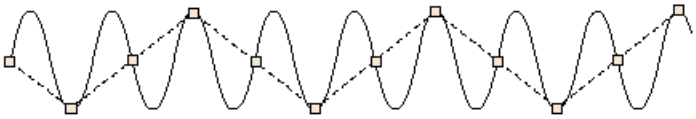
En la práctica se toma al menos entre 3 y 5 veces mayor que la frecuencia máxima. Como explicaremos en el próximo epígrafe, esto evita que las frecuencias superiores cercanas a la máxima produzcan ***aliasing***.

> Ej.: Las señales de audio recogidas y convertidas a señal eléctrica por un micrófono tienen (idealmente) componentes de hasta 20 kHz. Para poder reconstruirla, después de digitalizada, debe haber sido muestreada a una velocidad mayor de 40 kmuestras/s. (Los Compact Disc lo hacen a la velocidad normalizada de 44,8 kmuestras/s)

Aliasing (solapamiento)

EN EL DOMINIO DEL TIEMPO:

Señal muestreada a frecuencia superior a la de Nyquist.

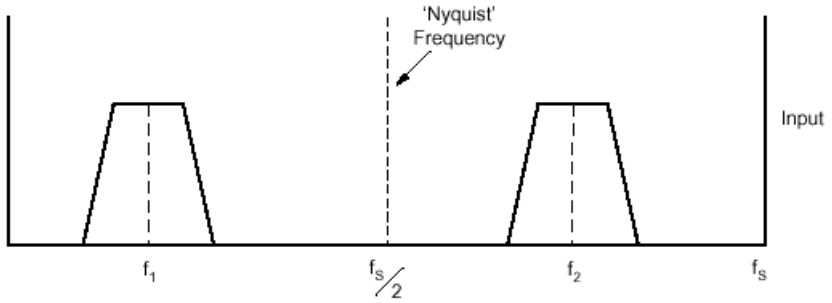


Muestreo a frecuencia inferior a la de Nyquist, aparece el *aliasing*: Al reconstruir la señal se obtiene otra frecuencia inferior.

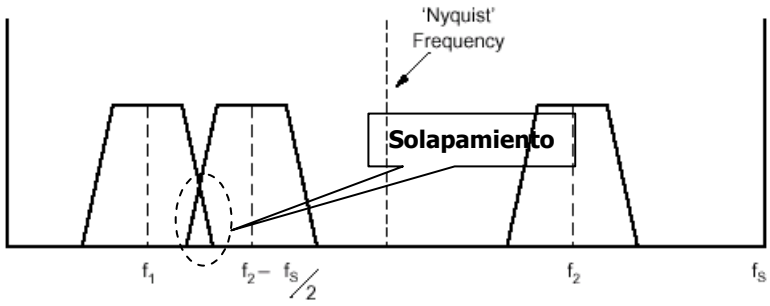
- ***PROPUESTO: En la imagen anterior, si la señal original fuera de 3 kHz, ¿Qué frecuencia observaríamos tras reconstruir la señal muestreada?***

EN EL DOMINIO DE LA FRECUENCIA:

Supongamos una señal de interés cuyo espectro está centrado en f_1 y otra interferente centrada en f_2 como pueden verse en la siguiente figura:



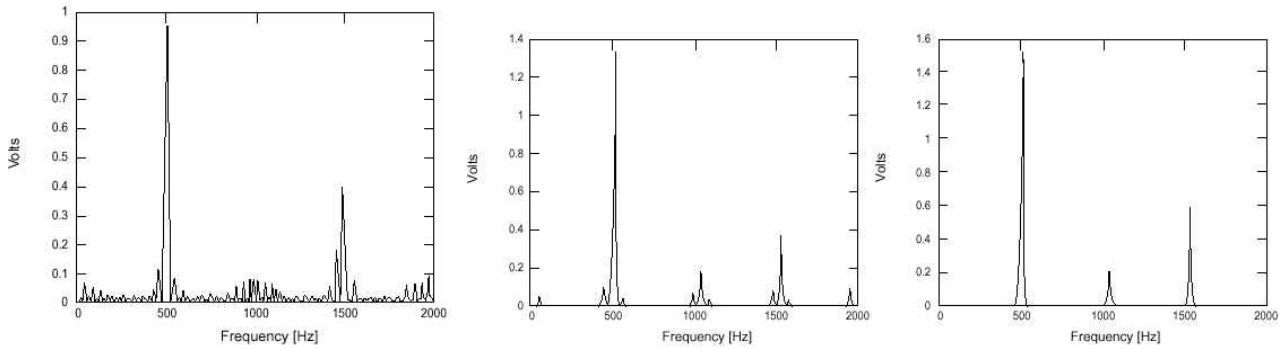
Cuando una señal de frecuencia f_x se muestrea a una velocidad f_s, su espectro aparece reflejado en $f_x - f_s/2$ como puede apreciarse en la siguiente figura:



En este caso, al muestrear la señal f_1, el espectro reflejado $f_1 - f_s/2$ queda a la izquierda del origen de frecuencias, por lo que no hay problema. Sin embargo al muestrear la señal f_2 su espectro reflejado $f_2 - f_s/2$, se solapa con f_1 confundiéndose con él. De esta forma, la señal f_2 interfiere con la señal útil f_1.

Ejemplo: señal cuadrada de 500 Hz:

Únicamente nos fijaremos en un ancho de banda limitado a 2 kHz, por lo que idealmente, además de la fundamental sólo debería existir un armónico a 1500 Hz.



- En la ***primera imagen*** la señal es muestreada a fs=4kHz y no se ha filtrado. Observamos *aliasing* de los armónicos superiores.
- En la ***segunda imagen***, fs=4kHz y previamente al muestreo se realizado un filtrado paso-bajo con frecuencia de corte de 2 kHz que ha eliminado muchos de los armónicos debidos al *aliasing*.
- En la ***tercera imagen***, además del filtrado se ha subido fs= 8 kHz eliminando los armónicos. El armónico de 1000 Hz es debido a la imperfección de la señal cuadrada.

El filtrado debe ser previo al muestreo (i.e. filtrado analógico). Después del muestreo, el *aliasing* no puede eliminarse; la señal ***alias*** no puede separarse de la señal útil mediante ningún tipo de filtro, pués 'cae' en su ancho de banda.

RESOLUCIÓN

Se refiere al incremento mínimo de tensión detectable, que coincidirá con el valor del bit menos significativo (***LSB***). Sin embargo, en tarjetas de adquisición de datos la resolución suele expresarse como el número de bits del conversor A/D.

Una resolución de 12 bits de un conversor A/D indica que es capaz de representar 4096 combinaciones binarias, es decir, 1 parte entre 4096 (2^{12}=4096).

LSB (LEAST SIGNIFICANT BIT)

Cuando hablamos de conversores A/D, el valor del bit menos significativo recibe el nombre de LSB.

$$1LSB = \frac{Rango}{nº\ cuentas} = \frac{Rango}{2^{nº bits}} \text{ (Voltios)}$$

Si tenemos en cuenta que antes del conversor A/D casi siempre hay un amplificador, debemos incluir su ganancia en el cálculo del ***LSB***.

$$1LSB = \frac{Rango}{Ganancia \cdot 2^{nº bits}}$$

donde el nº de bits se refiere al conversor A/D, y el rango a la diferencia entre el valor máximo y mínimo de tensión admitido en la entrada.

P.ej. 12 bits ➔ 10V en 4096 niveles ➔ 1 LSB = 2'44 mV (Ganancia=1)

- ***PROPUESTO: Hallar la resolución si el conversor fuera de 16 bits***

RANGO

Como puede verse en la siguiente tabla, el rango de entrada viene determinado por la ganancia seleccionada.

Tabla 1: Configuraciones de entrada para la serie PCI E de National Instruments

Range Configuration	Gain	Actual Input Range	Precision[1]
0 to +10 V	1.0	0 to +10 V	152.59 µV
	2.0	0 to +5 V	76.29 µV
	5.0[2]	0 to +2 V	30.52 µV
	10.0	0 to +1 V	15.26 µV
	20.0[2]	0 to +500 mV	7.63µV
	50.0[2]	0 to +200 mV	3.05 µV
	100.0	0 to 100 mV	1.53 µV
-10 to +10 V	1.0	–10 to +10 V	305.18 µV
	2.0	–5 to +5 V	152.59 µV
	5.0[2]	–2 to +2 V	61.04 µV
	10.0	–1 to +1 V	30.52 µV
	20.0[2]	–500 to +500 mV	15.26 µV
	50.0[2]	–200 to +200 mV	6.10 µV
	100.0	–100 to +100 mV	3.05 µV

[1] The value of 1 LSB of the 16-bit ADC; that is, the voltage increment corresponding to a change of one count in the ADC 16-bit count

[2] Not available on the PCI-MIO-16XE-50

Note: ***See Appendix A, Specifications, for absolute maximum ratings.***

- ***Donde dice*** precision ***debería decir*** resolution. ***La precisión máxima coincide con el valor de 1/2 LSB, pero no siempre es así debido a tensiones de offset, ruido, derivas térmicas, etc.***

- ***El producto*** Ganancia * Rango de entrada ***es constante:*** Rango configurado = Ganancia * Rango de entrada***.***

Configuración

El rango de las señales de entrada puede ser ***UNIPOLAR*** o ***BIPOLAR***

- UNIPOLAR: el rango de tensión de entrada está entre 0 y un valor positivo
- BIPOLAR: el rango de tensión de entrada está entre un valor negativo y un valor positivo

Se puede programar la polaridad y el rango de forma que cada canal tenga una configuración propia.

Hay que seleccionar el rango (Polaridad y ganancia) de forma que se ajuste al máximo al rango de la señal a medir obteniendo así la MAYOR RESOLUCIÓN posible. (1 LSB = $\frac{Rango}{Ganancia \cdot 2^N}$ Voltios)

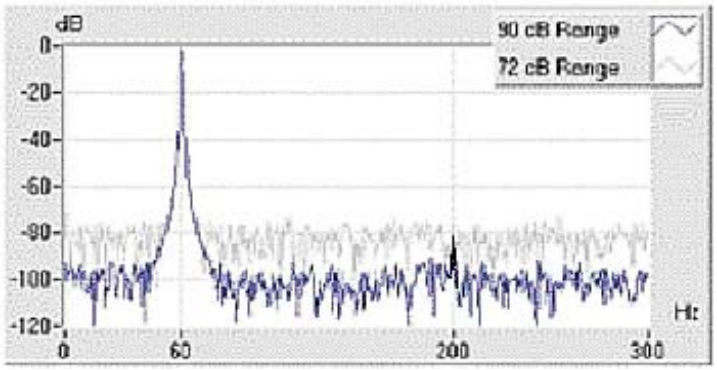
Rango dinámico

Expresa la diferencia máxima de magnitud que puede haber entre dos señales de entrada de forma que ambas puedan medirse. Suele expresarse en decibelios (***dB***).

Ejemplo: DAQ 12 bits, 10 V
Máx. tensión medible: 10 V
Mín. tensión medible: 1 LSB = $10V/2^{12}$ = 2'44 mV
Rango dinámico = $20 \log_{10} (10\ V/[10V/2^{12}]) = 20 \log_{10} (2^{12}) = 72$ dB

- Cada bit de resolución añade 6 dB (teóricos) al rango dinámico ($20 \log_{10} 2 = 6$)

Ejemplo: máquina rotativa
En una máquina rotativa, hay una frecuencia que determina la velocidad de rotación del eje y otras componentes pequeñas que pueden delatar un deterioro de los cojinetes. La relación entre las componentes pequeñas y la principal determina el mínimo rango dinámico. Las componentes pequeñas aumentan con el uso, de forma que un mayor rango dinámico del sistema de medida puede significar una detección más temprana.



$20 \cdot \log (1/2^{12}) = -72$ dB
$20 \cdot \log (1/2^{15}) = -90$ dB

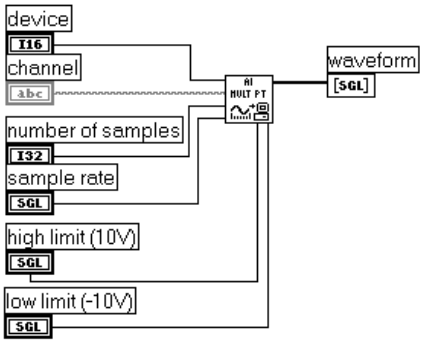
En la figura puede verse:
- En trazo claro con un rango dinámico de 72 dB (12 bits) una pequeña componente debida a una vibración de 200 Hz pasa desapercibida.
- En trazo oscuro con un rango dinámico de 90 dB (15 bits) se puede distinguir dicha componente.

PRÁCTICA 12: ADQUISICIÓN DE DATOS

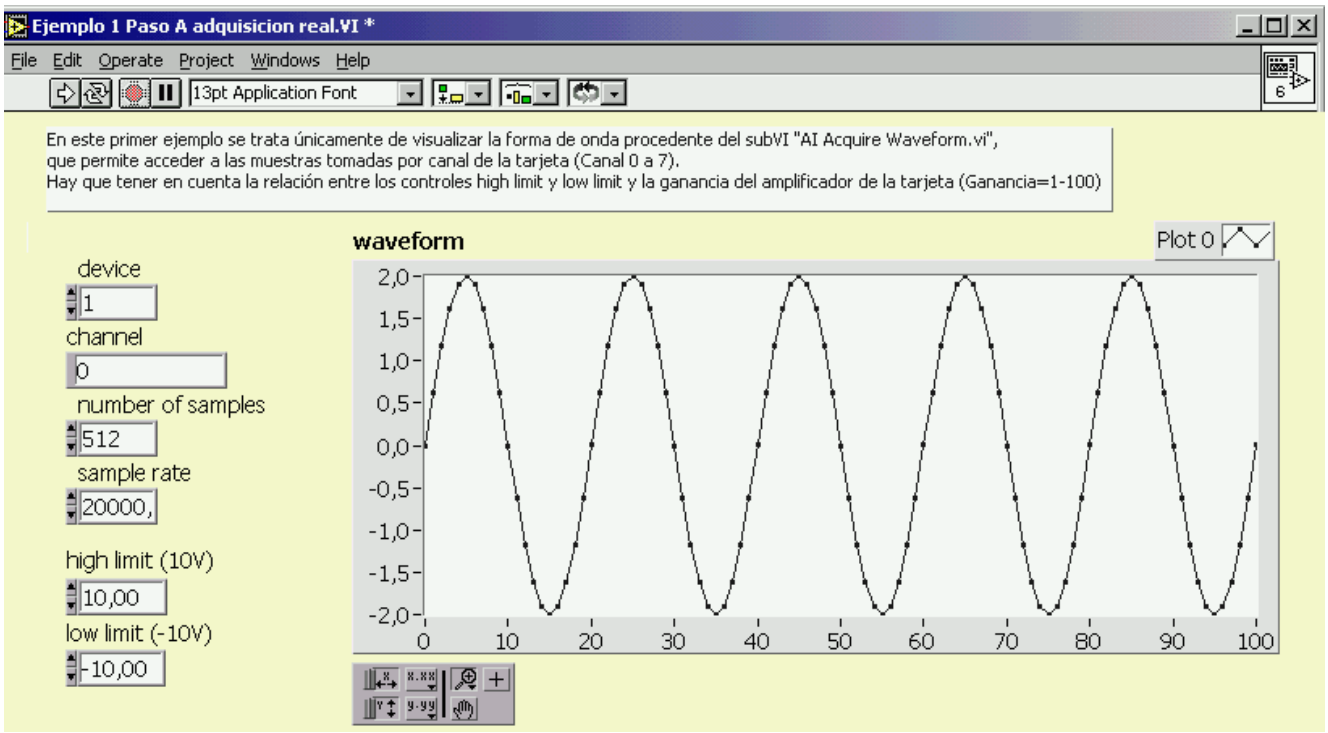
FUNCIONES RELACIONADAS CON LA ADQUISICIÓN.

LabVIEW posee una biblioteca completa para el **acceso al hardware** de adquisición de datos. Esta biblioteca está subdividida en funciones ***básicas*** y ***avanzadas*** según su complejidad.

PASO A: USO DE LA FUNCIÓN "AI ACQUIRE WAVEFORM.VI"



La función necesaria es:
Data Acquisition/Analog Input/AI Acquire waveform.VI.

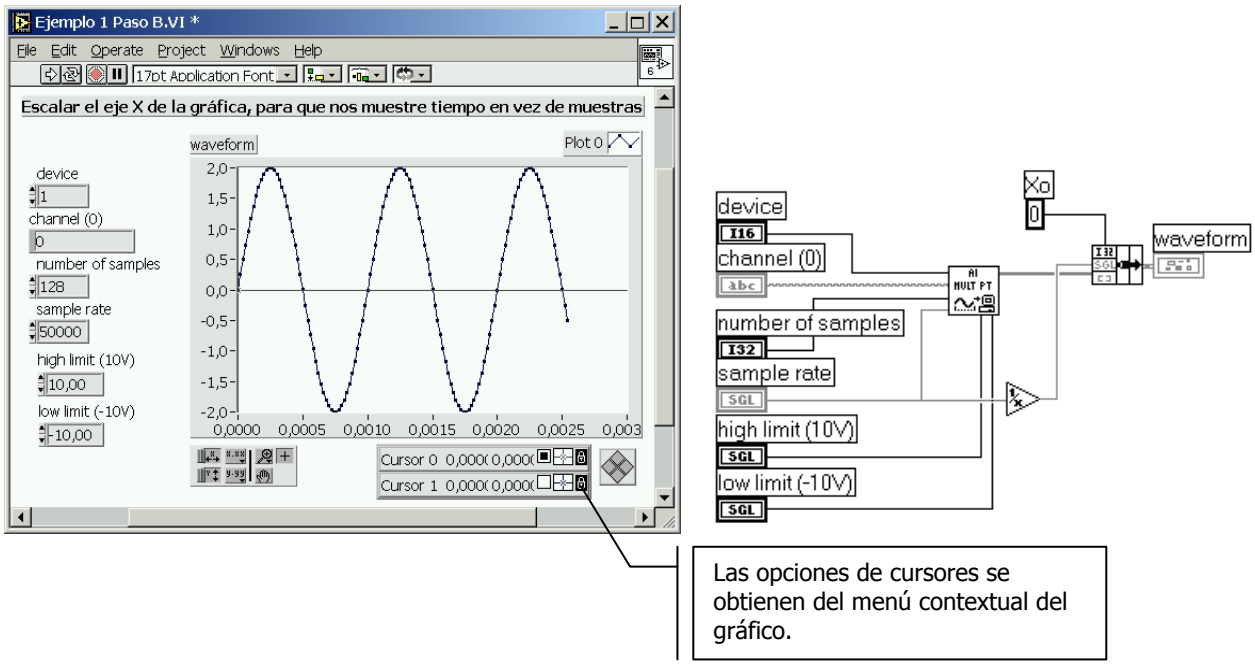


En la imagen se observa la señal obtenida si conectáramos, por ejemplo un generador de funciones al canal 0 de la tarjeta 1.

PASO B: ESCALADO DE TIEMPOS.

En esta segunda parte del ejercicio aprenderemos a escalar el eje X del gráfico, para que nos muestre tiempo en vez de muestras.

- Utilizar los cursores para determinar los tiempos y frecuencias de la señal visualizada.



MEDIDA DEL ERROR MEDIANTE CURSORES

OBJETIVO: Hallar el error de **frecuencia** del generador de funciones del laboratorio. Para ello:

1. Ajustar el generador de funciones a una señal sinusoidal.
2. Situar el mando del generador en la marca de 500 Hz.
3. Calcular manualmente el nº de muestras y la frecuencia de muestreo de forma que se vean 2 periodos completos con 10 puntos por periodo. (y así cumplir holgamente el teorema de Nyquist).
4. Ajustar el mando de amplitud del generador hasta que en el gráfico obtengamos 1 voltio de amplitud.
5. Ajustar *High limit* y *Low limit* (i.e. la ganancia) de la tarjeta para aprovechar al máximo su rango dinámico.
6. Hallar mediante los cursores del gráfico la frecuencia de la señal medida por nuestra tarjeta.
7. Rellenar la siguiente tabla:

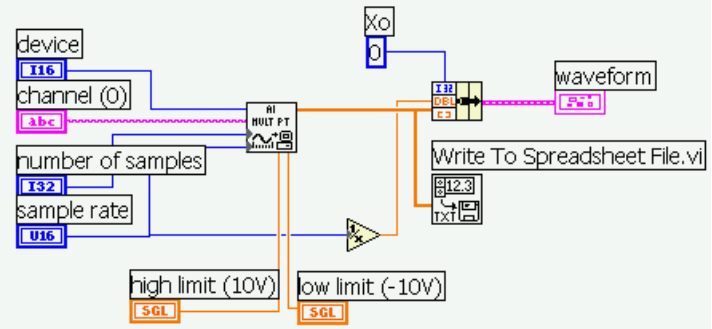
Ganancia seleccionada (buscar en especificaciones de la tarjeta)	
Nº de muestras	
Frecuencia de muestreo	
Medida según el generador de funciones	500 Hz
Medida en el gráfico	
Error absoluto	
Error relativo	

OBJETIVO: Hallar el error de **amplitud** del generador de funciones del laboratorio. Para ello:

1. Atenuar 20 dB la salida del generador de funciones pulsando el botón que incorpora para ello.
2. Ajustar el mando de amplitud del generador a fondo de escala (20 Vpp).
3. Ajustar *High limit* y *Low limit* (i.e. la ganancia) de la tarjeta para aprovechar al máximo su rango dinámico.
4. Hallar mediante los cursores del gráfico la amplitud de la señal medida por nuestra tarjeta.
5. Rellenar la siguiente tabla:

Ganancia seleccionada (buscar en especificaciones de la tarjeta)	
Nº de muestras	
Frecuencia de muestreo	
Medida según el generador de funciones (¡incluir los 20 dB!)	
Medida en el gráfico	
Error absoluto	
Error relativo	

OPCIONAL:
Guardar los datos adquiridos en un fichero con formato de hoja de cálculo:



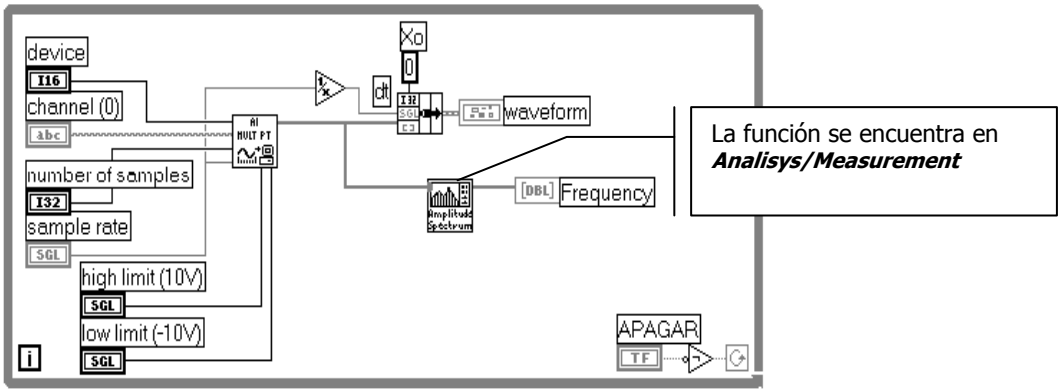
El VI que hemos añadido se encuentra en el grupo de funciones ***File I/O***

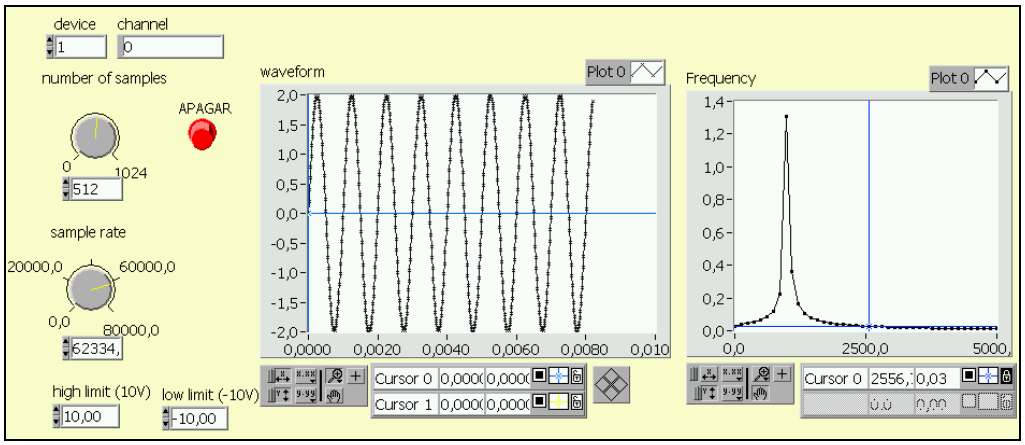
PASO C: ANÁLISIS EN FRECUENCIA

En esta tercera parte del ejercicio realizaremos las siguientes tareas:

1. Intercalar un subVI de análisis que calcula el espectro de amplitud. (Librería: ***Analisys/Measurement***)
2. Meter el grueso del programa dentro de un bucle ***While*** controlado por un botón de paro (Acción mecánica: ***Latch When Released***)
3. Cambiar el número de muestras y la velocidad de muestreo (sample rate) mediante controles tipo potenciómetro observando su efecto sobre la señal visualizada.
4. Observar el tipo de acción mecánica para el botón ***APAGAR***.

Ejemplo 1 Paso C



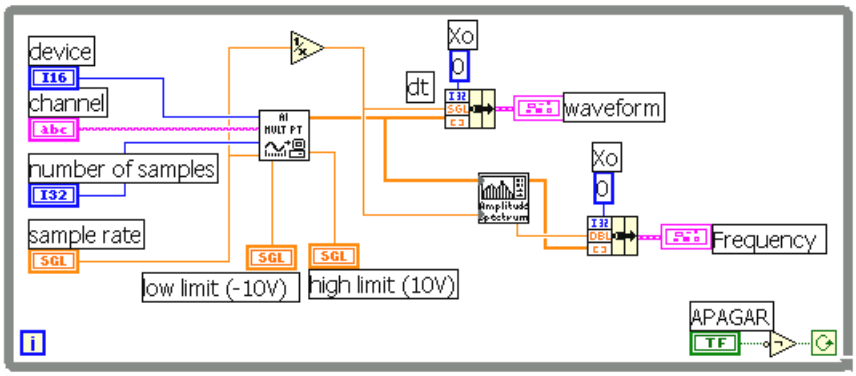


🛠 Rellenar la siguiente tabla con lo que se observa en cada gráfico al realizar las acciones descritas:

	Gráfico temporal	Gráfico espectral
Aumentar nº de muestras		
Disminuir nº de muestras		
Aumentar frecuencia de muestreo		
Disminuir frecuencia de muestreo		

PASO D: ESCALADO DE FRECUENCIA

De forma análoga a lo que se hizo para el eje de tiempos, debemos escalar el eje X del gráfico del espectro para que muestre la frecuencia en Hz. Observar las diferencias respecto al anterior diagrama:



MEDIDA MEDIANTE CURSORES

1. Generar una señal cuadrada de 100 Hz.
2. Ajustar la frecuencia de muestreo para cumplir el teorema de Nyquist hasta el 5º armónico.
3. Comprobar mediante los cursores del gráfico temporal que la frecuencia de la señal se corresponde con la que aparece en el espectro.
4. De la misma forma medir la amplitud

	amplitud	frecuencia
Medida en el gráfico temporal		
Medida en el espectro de frecuencias		
Error absoluto		
Error relativo		

RESOLUCIÓN ESPECTRAL

Se trata de averiguar de forma experimental la relación que define la resolución espectral ***Δf,*** para lo cual:

1. Seleccionar una señal sinusoidal de p.ej. 100 Hz
2. Mantener ***fs*** constante (cumpliendo Nyquist) y variar el nº de muestras ***N*** (P. ej. 128, 256, 512). ¿Qué ocurre con la separación entre los puntos del espectro, ***Δf*** ?.

fseñal	fmuestreo	Nº de muestras	Δf (aumenta ó disminuye)
100 Hz		128	
		256	
		512	
		1024	

3. Mantener el nº de muestras constante (P. ej. 256) y variar ***fs*** cumpliendo Nyquist. ¿Qué ocurre con el espectro?.

fseñal	Nº de muestras	fmuestreo	Δf (aumenta ó disminuye)
100 Hz	256		

4. A partir de los datos recogidos establecer la ecuación que define ***Δf*** en función de ***fs*** y ***N***. ¿Cuál de las tres expresiones es la correcta?

   Δf= N / fs Δf= fs / N Δf= fs * N
5. Comprobar numérica y experimentalmente la relación hallada para alguno de los casos anteriores:

fseñal	**Nº de muestras**	**fmuestreo**	**Δf**

ALIASING

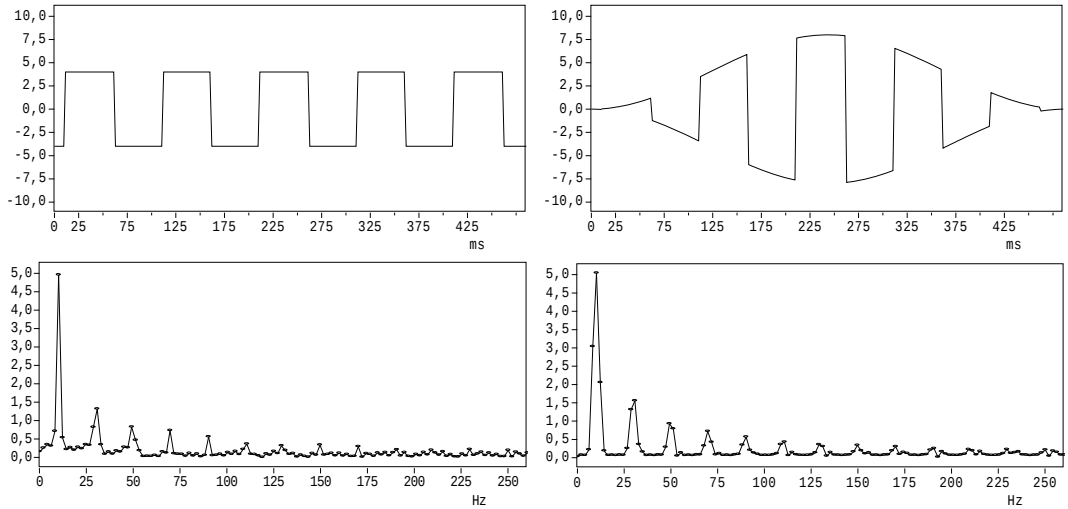
Rellenar la tabla siguiendo los pasos descritos:

Señal sinusoidal de 0'5 V@2000 Hz			
1. Inyectar una señal sinusoidal de 0'5 V@2000 Hz en el canal 0 de la tarjeta. **2.** Configurar la ganancia para obtener la máxima resolución. **3.** Muestrear la señal respetando el teorema de Nyquist. **4.** Medir la frecuencia de la señal (armónico único), comprobando que sea la esperada. **5.** Configurar el VI de forma que pueda rellenarse la tabla.	**Ganancia =**		¿_ _ _ _ _?
	Frec. muestreo (Mues./s)		**Armónico medido (Hz)**
	$f_s = f_{Nyquist}$		
	$f_s > f_{Nyquist}$		
	$f_s < f_{Nyquist}$		
CONCLUSIONES:			

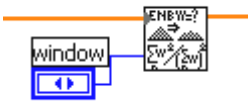
Observar el valor máximo del eje X del espectro. ¿Cuál es la relación del valor máximo con la frecuencia de muestreo?

PRÁCTICA 13: VENTANAS DE ALISADO

Con este ejercicio, trataremos de ver el efecto de realizar la FFT sobre un número **no** entero de periodos de la señal muestreada. Este efecto puede paliarse mediante el uso de ventanas de alisado previas a la FFT.



1. Insertar el VI ***Functions/Analysis/Measurement/Scaled Time Domain Window***

¿En que parte del diagrama debe insertarse?.	
	

2. Generar una señal sinusoidal de 500 Hz.
3. Muestrear a 5 kS/s

Calcular el nº de muestras para ver exactamente 3 periodos de señal

4. Ajustar el nº de muestras a lo calculado para ver exactamente 3 periodos de señal.
5. Observar el espectro de amplitud, especialmente los alrededores del armónico fundamental.
6. Ajustar el nº de muestras que permita ver exactamente 2'5 periodos de señal.
7. Observar el espectro de amplitud, especialmente los alrededores del armónico fundamental.
8. Activar la ventana tipo "Hamming" y repetir los pasos anteriores.
9. Conclusiones:

¿Qué sucede cuando el número de muestras por periodo es un número entero?.
¿Qué sucede cuando el número de muestras por periodo NO es un número entero?.

PRÁCTICA 14: FILTRADO DIGITAL

14.1.- PROMEDIADO

El proceso más evidente de filtrado es el promediado. La relación Señal/Ruido (potencia) que se obtiene mediante esta técnica es:

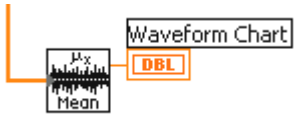
$$\frac{N_{out}}{N_{in}} = \frac{1}{\sqrt{nº\,muestras}}$$

Por ejemplo un promediado de 4 muestras reduce a la mitad la potencia de ruido en la salida. Observar que para reducir a la cuarta parte dicha relacion tendríamos que usar un promediado de 16 muestras

14.1.1.- Eliminar interferencia de la red eléctrica mediante promediado

Cuando tenemos una señal cuasi continua a la que se ha añadido una interferencia como la de la red eléctrica, podemos aprovechar que el valor medio de una sinusoide es cero para eliminarla. Si tomamos un nº de muestras tal que completen un ciclo de la sinusoide, su valor medio será cero. Se trata de modificar nuestro VI para que realice dicha tarea:

1. Verificar que la tarjeta se encuentra configurada en modo DIFF (diferencial)
2. Cortocircuitar las entradas del PGIA (Así la tensión diferencial será cero, pero no la tensión de modo común).
3. Fijar una velocidad de muestreo suficiente para la interferencia de la red eléctrica.
4. Fijar la ganancia de la tarjeta al rango de la señal para aprovechar al máximo la resolución.
5. Verificar mediante el VI anterior que la frecuencia de la señal interferente se corresponde con la esperada.
6. Calcular el **nº de puntos, N**, del promediado que debemos realizar para eliminar dicha interferencia.
7. Modificar el VI anterior para que realice este cometido añadiendo los siguientes elementos:



8. Fijar el nº de muestras del VI al nº de puntos calculado, y comprobar el resultado.
9. Rellenar la siguiente tabla con la ecuación que determina el nº de muestras, N, y los datos obtenidos:

Frec. muestreo	$N = \frac{¿.....?}{¿.....?}$	**Observaciones**
f_1 = 400 Hz		
f_2 =		

14.1.2.- Eliminar ruido y rizado de la medida de una señal DC mediante promediado

Introducción:

Cuando deseamos medir una señal DC, ésta puede presentar un rizado, además del ruido añadido por su circuitería y las interferencias de sistemas cercanos. Si tomamos un nº de muestras suficiente y hallamos su valor medio, nos acercaremos a su valor DC.

Objetivos:

- Medir adecuadamente una señal DC flotante.
- Utilizar el promediado como método de filtrado del ruido y rizado presentes en una señal DC.

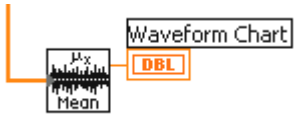
Material:

- Fuente de señal: Salida fija de 5 volts de la fuente de alimentación Promax FAC-363B.

Realización:

ANTES DE CONECTAR

1. Tras acudir a la documentación de la tarjeta, seleccionar la configuración adecuada, (DIFF vs NRSE, DIFF vs RSE), Ganancia.
2. Dibujar en papel el circuito donde consten la fuente, el PGIA y la conexión con los números de terminales usados.
3. Mostrar dicho circuito al profesor para que lo verifique.
4. Modificar el programa para que realice el promediado mostrando el valor de la tensión en un gráfico tipo *chart*.



5. Fijar el nº de muestras de forma que la relación señal/ruido (potencia) se reduzca a la octava parte (-9 dB).
6. Conectar a la salida fija de 5 volts de la fuente de alimentación Promax FAC-363B
7. Rellenar la siguiente tabla con los datos obtenidos:

Error absoluto	Error relativo	Observaciones

> ***OPCIONAL : Probar a medir en configuración diferencial una señal flotante procedente, por ejemplo, de la fuente de alimentación.***

14.2.- RUIDO

VISUALIZACIÓN DEL RUIDO INHERENTE A LA TARJETA DAQ:

1. Verificar que la tarjeta se encuentra configurada en modo DIFF (diferencial)
2. Conectar las entradas + y – de un canal cualquiera de la tarjeta a la masa del sistema AIGND.
3. Configurar la tarjeta a la más alta velocidad de muestreo (≈ 200 kS/s, dependiendo del modelo de tarjeta).
4. Ajustar la ganancia al máximo.
5. Cerciorarse de que estamos visualizando una señal aleatoria (i.e. ruido)
6. Aumentar paulatinamente el nº de muestras, rellenando la siguiente tabla:

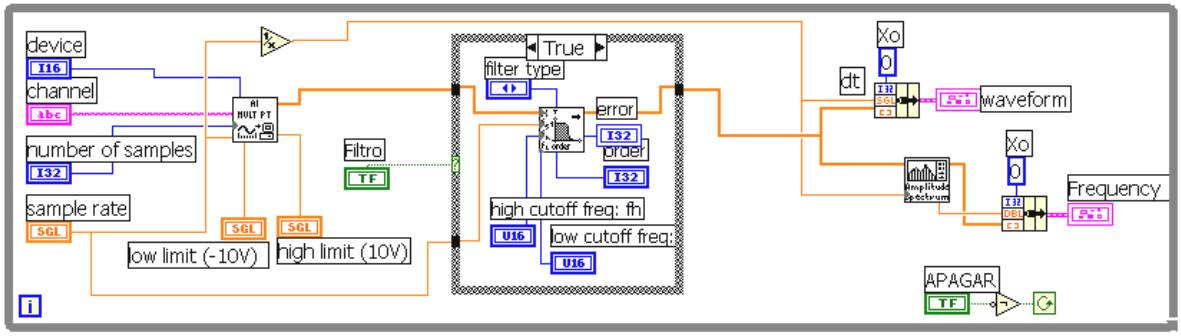
	N	Observaciones sobre el espectro
fs = _____	1000	
	5000	
	10000	
¿Cómo se distribuyen los armónicos en el eje de frecuencia?:		
¿A qué /quiénes imputarías la presencia de esta señal aleatoria?.		

14.3.- FILTROS DIGITALES

AÑADIR FILTRADO DIGITAL DE BUTTERWORTH

Vamos a intercalar un filtro entre el array de señal y los gráficos de forma que podamos filtrar varios armónicos de una señal cuadrada (distorsionar). Para ello:

1. Crear una estructura ***case*** que permita al usuario elegir si desea filtrar o no la señal mediante un botón.
2. Dentro del caso ***True*** incorporar un filtro Butterworth (se encuentra en el grupo de funciones ***Analysis/Filters/Butterworth filter.vi***)
3. Crear automáticamente los controles necesarios para configurar el filtro: ***tipo, orden, frecuencia superior de corte, frecuencia inferior de corte***. Debemos obtener algo parecido a la siguiente figura.



4. Generar una señal cuadrada de 1 kHz.
5. Aislar el 1er armónico mediante el filtro dispuesto.

Orden del filtro	Nº de muestras	fmuestreo	Δf	f teórica del 1er armónico	f medida del 1er armónico

En el filtro debe cumplirse 0 < fcorte ≤ fs/2, de lo contrario dará un error. Si, según Nyquist sólo pueden muestrearse señales por debajo de fs/2, el filtro sólo funcionará hasta esa frecuencia.

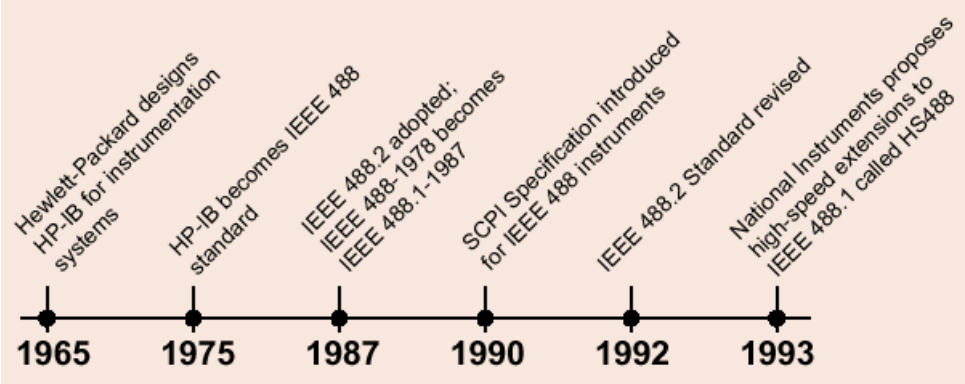
Describir la influencia del orden del filtro en los resultados obtenidos:

4.- BUS GPIB.

INTRODUCCIÓN AL BUS GPIB

HISTORIA

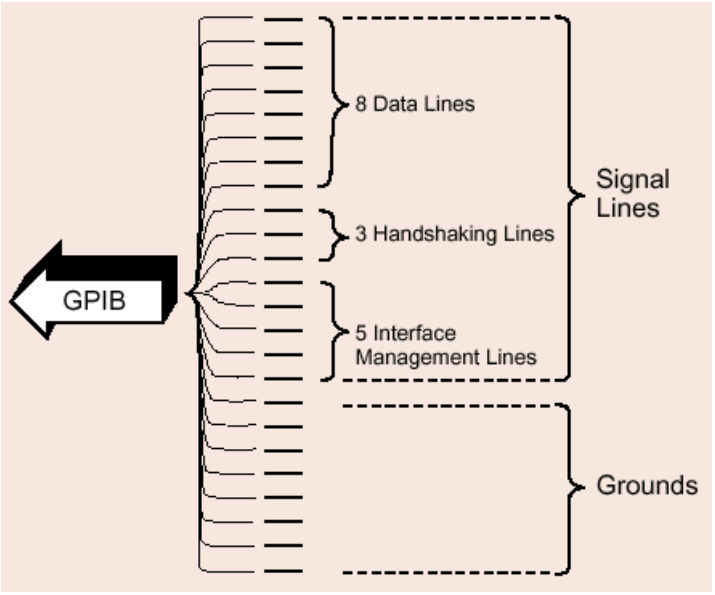
- En 1965 la empresa Hewlett Packard crea la interfaz de bus HP-IB (Hewlett Packard Interface Bus) para conectar instrumentos programables a sus ordenadores.
- Debido a sus ventajas se populariza de tal forma que en 1975 se estandariza como IEEE 488-1975 (Aunque es más conocido como GPIB -General Purpose Interface Bus-)
- En 1987 evoluciona dando paso al ANSI/IEEE 488.2, por lo que el anterior IEEE 488-1975 pasa a denominarse ANSI/IEEE 488.1



National Instruments™

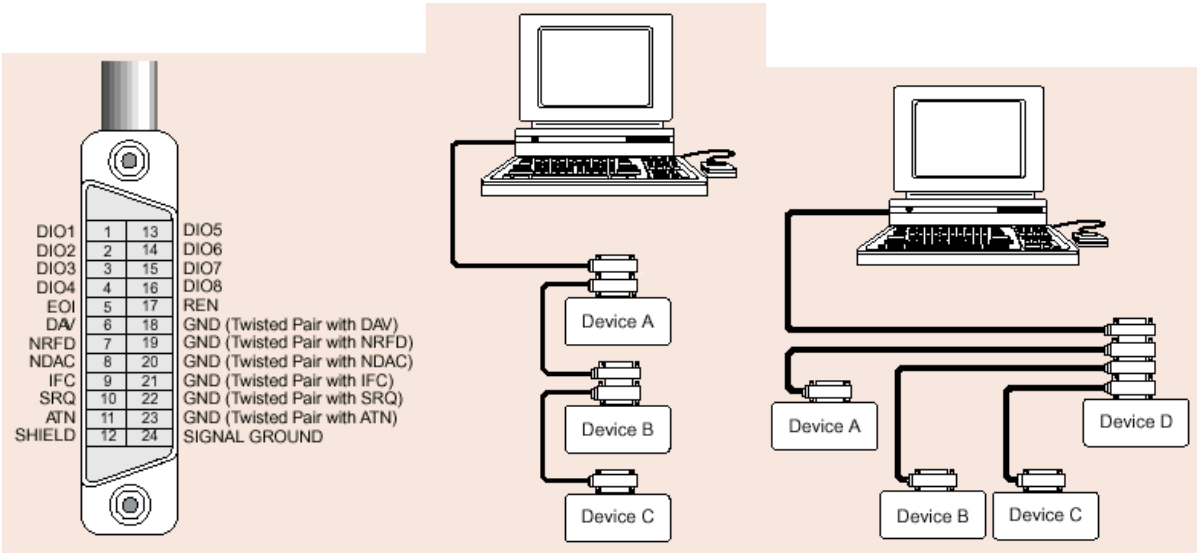
CARACTERÍSTICAS MECÁNICAS Y ELÉCTRICAS

Se trata de un bus paralelo, formado por 8 líneas de datos, 3 líneas de protocolo, 5 de gestión, 7 de masa y 1 apantallamiento. Tenemos por tanto un cable de 24 conductores.



National Instruments™

Permite la conexión de dispositivos en estrella, o en linea:



National Instruments™

- Utiliza niveles TTL y lógica negada.
- Velocidades de transferencia de hasta 1 MB/s.
- Hasta 15 dispositivos conectados al bus.
- Separación máxima de 4 m entre dos dispositivos cualesquiera del bus. Longitud total del cable de hasta 20 m

FUNCIONAMIENTO

Los dispositivos conectados al bus se comunican entre ellos mediante mensajes

Tipos de mensajes

- ***Dependientes del dispositivo***: Llamados datos a secas, contienen información específica de los dispositivos como pueden ser resultados de una medida, estado de un aparato, instrucciones de configuración, etc.
- ***De gestión***: También llamados mensajes de órden, realizan funciones como inicializar el bus, direccionar dispositivos, etc.

Roles de los dispositivos

Los dispositivos conectados al bus pueden adoptar uno o varios de los roles definidos: Hablante, escuchador y controlador.

ESCUCHADOR (LISTENER)

Capaz de recibir datos de la interfaz cuando esté direccionado (habilitado) por el controlador. Puede haber hasta 14 escuchadores activos simultaneamente en el bus. Dispositivos escuchadores pueden ser por ejemplo una impresora, un generador de funciones, etc..

HABLANTE (TALKER)

Un hablante, cuando se le ha direccionado, envía mensajes a uno o varios escuchadores que reciben los datos. Por ejemplo un osciloscopio puede actuar como hablante y como escuchador. Sólo puede haber un hablante activo sobre la interfaz.

CONTROLADOR

Gestiona el flujo de información en el bus enviando órdenes a todos los dispositivos. El controlador es capaz de direccionar (habilitar) a un hablante que quiera enviar un mensaje a varios escuchadores, permitiendo una operación de transferencia de datos. Él mismo puede ejercer de emisor o receptor.

Se podría configurar un bus sin dispositivo controlador y en el que hubiera un dispositivo con capacidad únicamente de hablante y varios con capacidad únicamente de escucha.

La función de controlador la suele realizar un dispositivo conectado al ordenador (Tarjeta PCI, PCMCIA, etc.) que habitualmente tiene además capacidad de hablante y escuchador.
En sistemas con varios controladores sólo uno puede estar actuando como tal. El controlador activo puede pasar el mando del bus a otro controlador que se halle inactivo.

Lineas según su función

LINEAS DE DATOS

Son 8 y se denominan ***DIO1-DIO8***

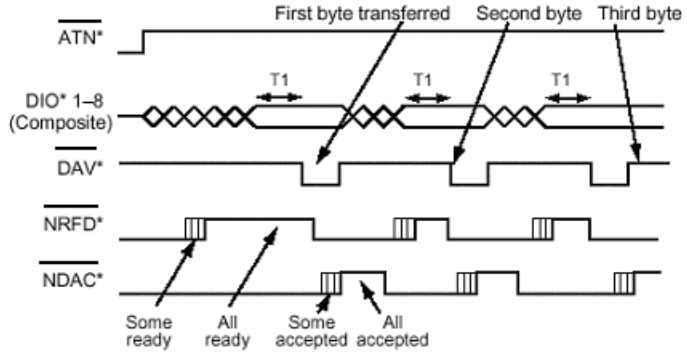
- El estado de la linea de gestión ***ATN*** (se verá más adelante) determina si la información presente en el bus de datos corresponde a un dato o a una órden.
- Todas las órdenes y la mayoría de datos utilizan el código ASCII de 7 bits, de forma que la linea ***DIO8*** se utiliza como paridad.

LINEAS DE PROTOCOLO

Estas tres lineas controlan de forma asíncrona la transferencia de bytes de mensajes entre dispositivos, garantizando que se realiza sin errores.

- ***DAV*** (DAta Valid):. Es activada por el controlador cuando envía órdenes y por el hablante cuando envía mensajes de datos. Indica cuando la señales de datos son estables (válidas) de forma que puedan ser interpretadas de forma fiable por los escuchadores.
- ***NRFD*** (Not Ready For Data): Un escuchador activa esta linea cuando no está preparado para la recepción de datos. Tendrá valor *falso* cuando todos los receptores direccionados estén listos para recibir datos. Se realiza la función ***OR cableada*** de las salidas ***NRFD*** de lo escuchadores direccionados. Siendo necesario que todos estén preparados (todas las salidas ***NRFD*** a *falso*) para que la línea ***NRFD*** se ponga a *falso*.
- ***NDAC*** (No Data ACcepted): En estado *verdadero* indica que algún escuchador direccionado no ha aceptado todavía los datos enviados. Un estado *falso* indica que todos los receptores activos han aceptado los datos. Nuevamente se realiza la función *OR cableada* de todas las salidas ***NDAC*** de los receptores activos.

En la siguiente figura puede verse la secuencia de control de transferencia de datos o ***handshake***.



National Instruments Corporation

LINEAS DE GESTIÓN

Estas cinco lineas gestionan el flujo de datos e información del bus.

- ***ATN*** (ATtentioN): El controlador activa ***ATN*** a *verdadero* cuando usa las lineas de datos para enviar órdenes y la pone a *falso* para permitir que un hablante envíe datos.
- ***IFC*** (Interface Clear): El controlador activa esta linea para inicializar el bus interrumpiendo el proceso que se estaba realizando. Se deshabilita al hablante y a los escuchadores activos, quedando todos inactivos. El controlador asume el mando del bus. Todos los dispositivos deben responder a esta línea en cualquier instante.

- ***REN*** (Remote Enabled): El controlador activa esta linea para poner los dispositivos direccionados en modo de programación remota. Cuando no está activada los dispositivos se encuentran en modo de control local.
- ***SQR*** (Service Request): Cualquier dispositivo puede activar esta linea para pedir servicio al controlador.
- ***EOI*** (End Or Identify): Tiene dos funciones; el hablante activa esta linea para indicar el último byte de un dato a los receptores activos. Con ATN a "1", la activación de esta línea indica que el controlador realiza un sondeo paralelo.

IEEE 488-2

- Viene a paliar los problemas que arrastra el estándar desde el veterano IEEE 488.1.
- Se mantiene la compatibilidad con IEEE 488.1, pero los beneficios del nuevo estándar sólo se obtienen cuando se tiene un sistema totalmente compatible IEEE 488.2
- Normaliza el lenguaje de programación de los dispositivos incorporanto el estándar SCPI.

SCPI (Standar Commands for Programmable Intruments)

Una de las cosas que no define el estándar IEEE 488.1 es la estructura y sintaxis del lenguaje que se usa para programar los dispositivos, lo que provoca que p. ej. un programa elaborado para un osciloscopio no funcione con un osciloscopio de otra marca. Esto aumenta el trabajo de los programadores y por tanto los costes de desarrollo y mantenimiento. Podemos comparar el problema con el caso de las lineas telefónicas: puedo establecer una comunicación con China por que las lineas son compatibles, pero si no conozco el idioma, no lograré que el hablante y el escuchador se entiendan. SCPI viene a solventar esa dificultad idiomática entre instrumentos definiendo un estructura y sintaxis común del lenguaje.

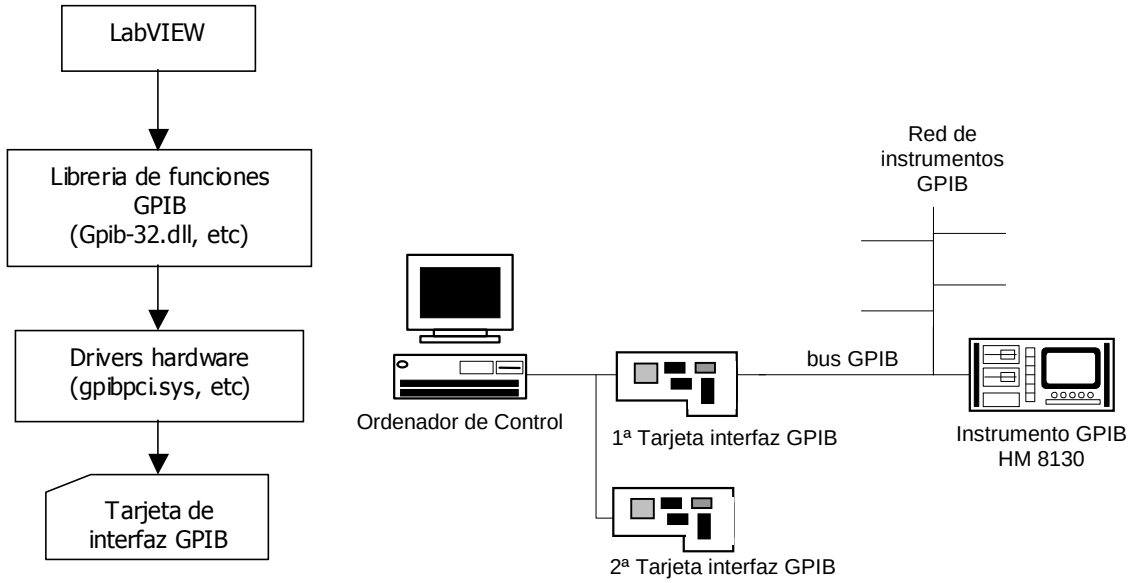
HS488

Es una propuesta de la compañia National Instruments para la mejora de las velocidades de transmisión de la norma 488. Entre dos dispositivos compatibles HS488 separados por 2 metros de cable se pueden alcanzar de hasta 8 MB/s. En un sistema con 15 dispositivos y 15 metros de cable la velocidad puede alcanzar 1’5 MB/s

PROGRAMACIÓN GPIB CON LABVIEW

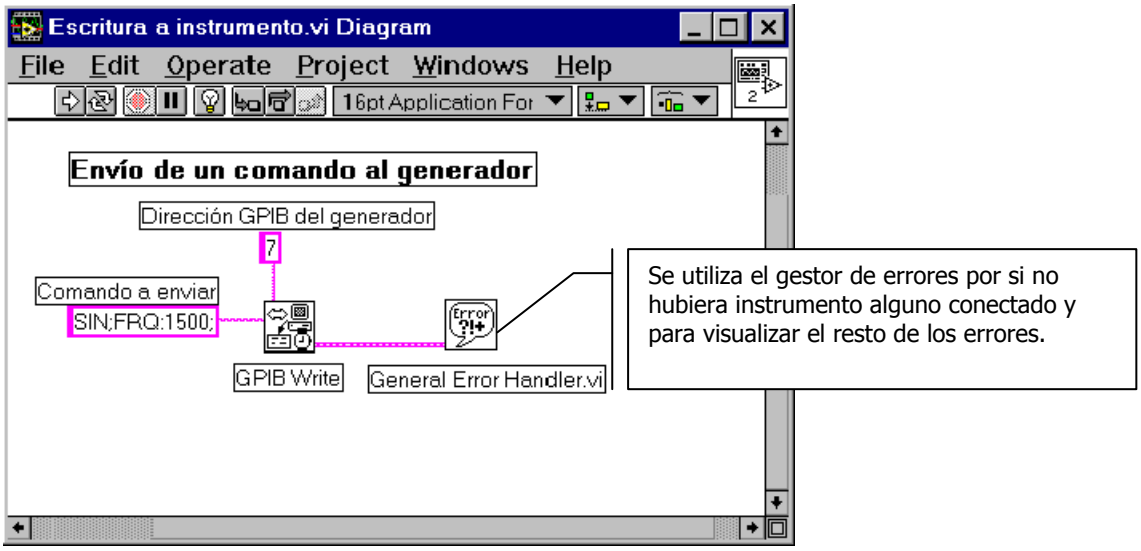
CONEXIÓN LABVIEW-DISPOSITIVO GPIB

Las siguientes figuras ilustran la conexión que permite controlar un instrumento GPIB mediante el software LabVIEW cargado en nuestro ordenador.



ENVÍO DE ÓRDENES

En este apartado veremos como enviar ordenes por el bus GPIB al generador de funciones Hameg HM-8130 mediante LabView.
Las funciones relativas al bus GPIB se encuentran en *Functions/ Instrument I-O/ GPIB*
Debemos conocer previamente la dirección del bus asignada a nuestro aparato. Podemos verla directamente según la la configuración de los microinterruptores que se encuentran es su parte trasera. Además nos lo muestra en su display cada vez que encendemos el aparato.



Entre las órdenes que admite el generador de funciones HM-8130 tenemos por ejemplo:
SIN Genera a la salida la onda sinusoidal de los valores seleccionados
OT0 Desactiva y ***OT1*** activa la salida de señal.
OF0 Desactiva la tensión de Offset en la salida y ***OF1*** la activa.
DFR Visualiza la frecuencia.
DAM Visualiza la amplitud
LK1 Bloquea la botonera del generador para impedir su manejo en modo local. ***LK0*** lo desbloquea.

Otras órdenes que precisan adjuntar un dato son por ejemplo:

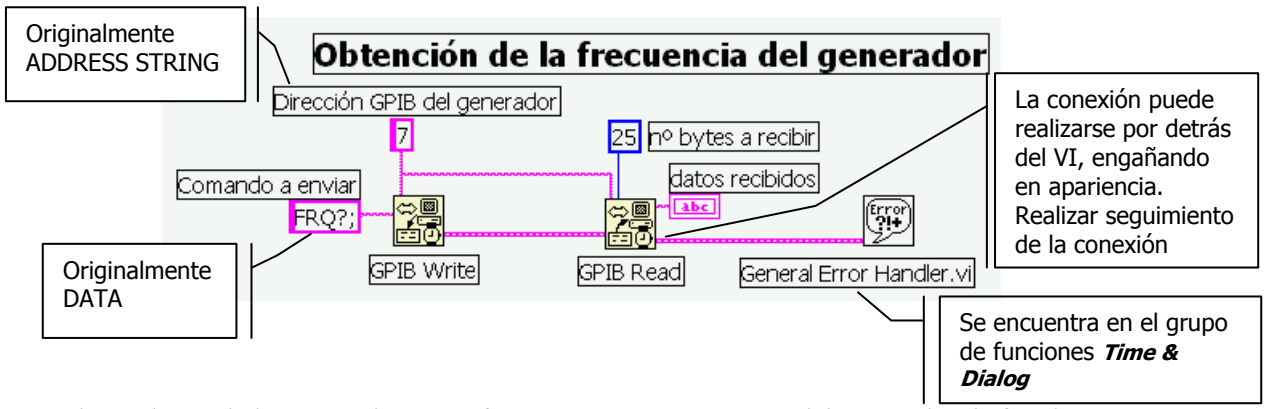
FRQ:dato Sitúa la frecuencia de la señal en el valor de 'dato' (Hz).
AMP:dato Sitúa la amplitud de salida según 'dato' (Voltios).
OFS:dato Sitúa la tensión de Offset al valor expresado por 'dato' (Voltios).

Las órdenes del generador pueden concatenarse para formar una única cadena de órdenes y enviarse a la vez. Cada órden debe acabar con punto y coma "***;***".Por ejemplo:

LK1;TRM;RMP;FRQ:0;AMP:2,000E-2;OT1;OF0;

RECEPCIÓN DE RESPUESTAS

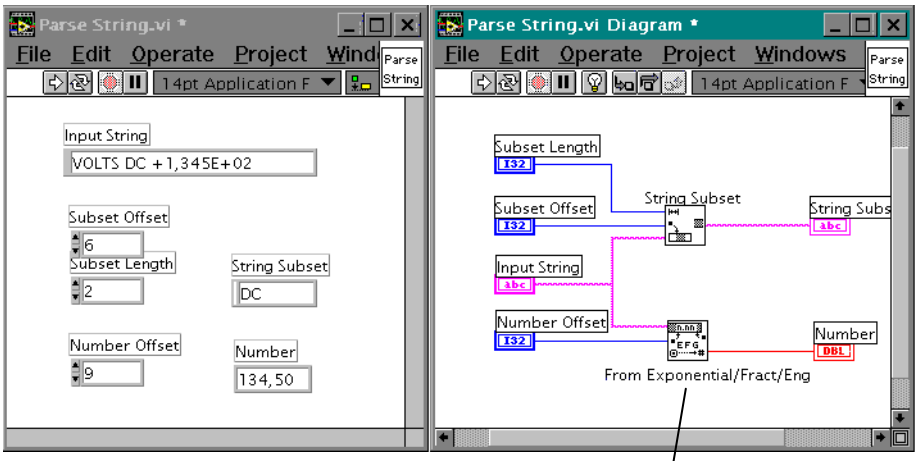
La comunicación con los instrumentos implica la recepción de respuestas tras el envío de órdenes de interrogación. Por ejemplo para saber la frecuencia actual de la señal generada.
El subVI ***General Error Handler*** mostrará en pantalla los errores producidos en la comunicación pro el bus en caso de producirse. y asegura (al unir la salida de la acción de escritura con la de lectura) la correcta secuencia; no hay ninguna duda de que primero se producirá la escritura de la órden de interrogación, en espera de la respuesta y posteriormente la lectura de la misma.



Entre las órdenes de interrogación que ofrecen respuesta por parte del generador de funciones están:
FRQ? Devuelve la frecuencia actual
AMP? Voltaje actual a la salida
OFS? Tensión de Offset.
ID? Identificación del aparato.
VER? Versión del equipo.
STA? Estado del equipo.

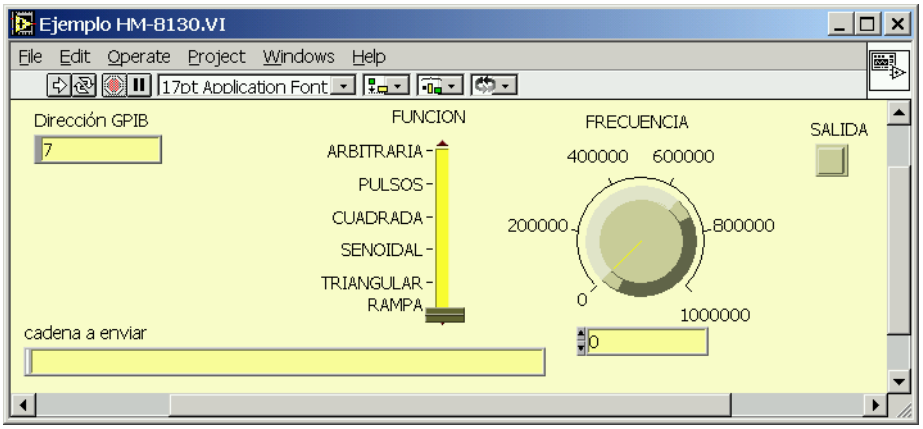
OBTENCIÓN DE DATOS DE LA CADENA DE RESPUESTA DE UN INSTRUMENTO:

El instrumento devolverá como respuesta a nuestra interrogación una cadena de caracteres que deberemos tratar para obtener la información de interés. Para ello trocearemos la respuesta, en cadenas de caracteres y números, para su posterior utilización.

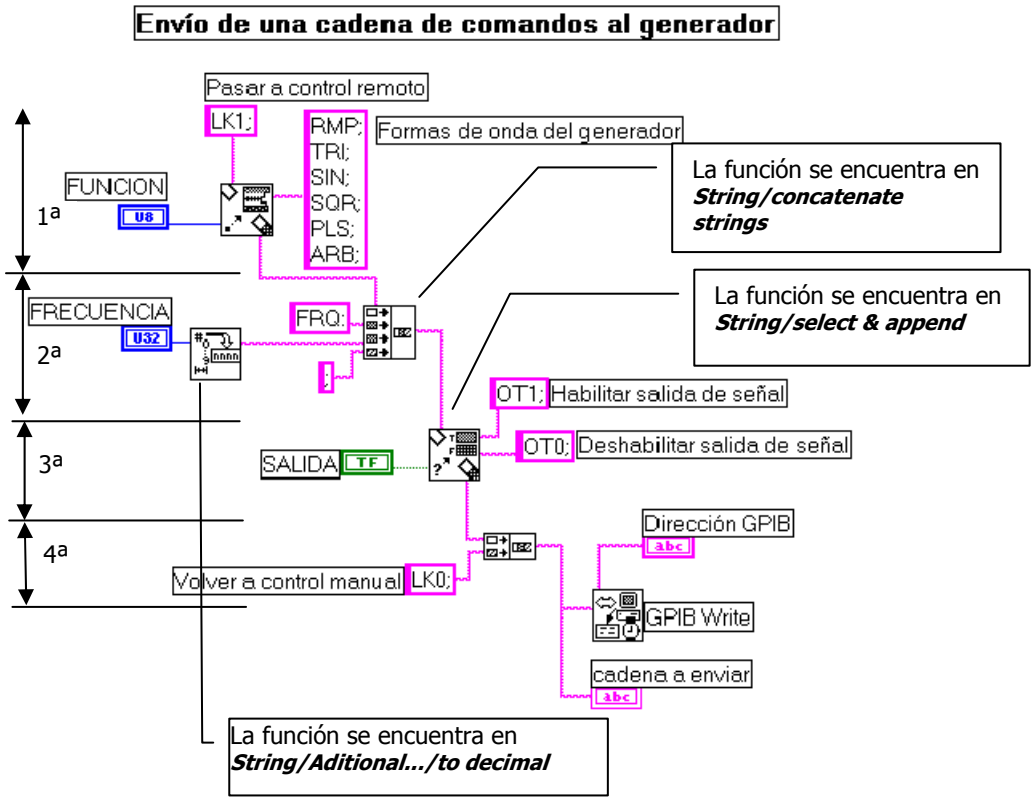


La función se encuentra en
FUNTIONS/strings/additional string to number

En general es interesantre para el usuario que la interfaz de nuestro VI se parezca lo más posible al panel frontal del instrumento real. Colocaremos mandos para todas las funciones que necesitemos controlar.



Por razones didácticas, hemos añadido un visualizador de la cadena que se enviará al instrumento.
El diagrama de bloques que debemos implementar es el siguiente:



Realizar el programa por partes, visualizando el resultado de cada una de ellas (de 1ª a 4ª) según se incorporan nuevas funciones. Por último añadir el sub VI de escritura en el bus GPIB.

APLICACIONES GPIB EN RED

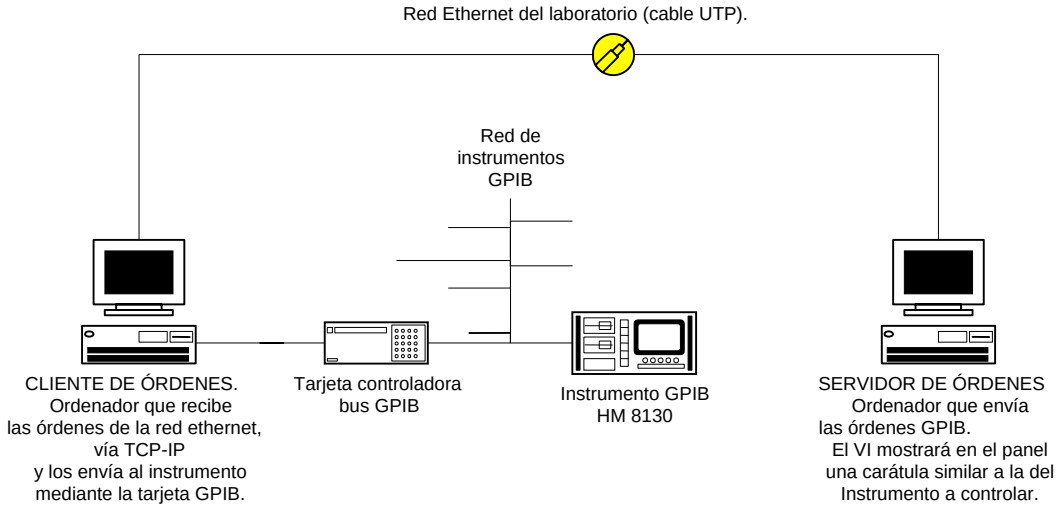
Podemos ver otras aplicaciones como son las relativas al control de instrumentación no directamente conectada a nuestro ordenador.

SERVIDOR REMOTO DE ÓRDENES GPIB

Por ejemplo es posible controlar un dispositivo GPIB desde cualquier puesto de una red de ordenadores. Un ordenador contiene la tarjeta de interfaz GPIB, pero el control lo podemos realizar desde otro ordenador, que sería el que contenga el programa de aplicación.

Los datos llegarían al aparato a controlar mediante el ordenador que tiene la tarjeta de bus, que actuaría como un servidor de datos; estos serían enviados por el ordenador que tiene el programa de control y que actuaría como un cliente.

Este intercambio de datos se hace mediante la tecnología cliente-servidor dentro de las redes de ordenadores, y permite la conexión remota, incluso vía telefónica entre distintos equipos de distintas características siempre y cuando cumplan con el protocolo de comunicación.
Conexión para manejar el generador de funciones con órdenes GPIB desde otro PC. El ordenador que no tiene la tarjeta controladora es el que genera las órdenes de control. Ambos ordenadores se comunican mediante la red ethernet en el protocolo TCP-IP.



APLICACIONES DE DISTRIBUCIÓN DE SEÑAL EN RED

Cliente de la adquisición de señal.

Uno de los ordenadores, el que tiene la tarjeta de adquisición, manda a los otros, mediante TCP-IP por la red ethernet, la señal capturada.