

Práctica 1

Introducción a la instrumentación virtual. El entorno de trabajo LabVIEW 5.0

Objetivo

El objetivo de esta práctica es familiarizarse con el entorno de trabajo LabVIEW. Para ello se diseñará paso a paso un programa sencillo, un generador de señal senoidal.

Introducción a la instrumentación virtual. Control de instrumentos desde el ordenador. Automatización de medidas

Muchas veces la realización de una medida requiere la intervención de varios instrumentos. Unos generan estímulos sobre el dispositivo que se pretende medir y otros recogen la respuesta a estos estímulos. Este conjunto de instrumentos que hace posible la realización de la medida recibe el nombre de sistema de instrumentación. Todo sistema de instrumentación consta de unos instrumentos, un sistema de interconexión de estos instrumentos y un controlador inteligente que gestiona el funcionamiento de todo el sistema y da las órdenes para que una medida se realice correctamente.

La utilización manual de instrumentos para realizar medidas es prácticamente un hecho aislado. Sólo en los procesos de investigación y desarrollo de nuevos prototipos o en entornos docentes es una tarea habitual. A nivel industrial, las medidas para el control de un determinado proceso, las pruebas funcionales sobre un equipo o el control de calidad de la producción se realizan de manera automática. La automatización de las medidas requiere que los instrumentos gocen de un cierto grado de inteligencia para que puedan ser gobernados por un controlador que se comunica con los instrumentos a través de un BUS de instrumentación (GPIB, VXI, EIA-232, RS-485...). La figura 1.1 muestra un sistema de instrumentación virtual.

Los inicios de la instrumentación controlable desde el ordenador, y de hecho de los sistemas de instrumentación, se sitúan a mediados de los años 60 cuando Hewlett Packard, desarrolló su bus para instrumentación HP-IB (Hewlett Packard Interface Bus) que permitía conectar su gama de instrumentos programables a un ordenador. Esta interfase ganó rápidamente gran popularidad y en 1975 fue aceptada como un estándar: el IEEE-488. Desde aquellos días hasta ahora el estándar ha

sufrido varias modificaciones y el bus GPIB (acrónimo de General Purpose Interface Bus, por el que se le conoce habitualmente) se ha convertido en uno de los más populares en el campo de la instrumentación programable

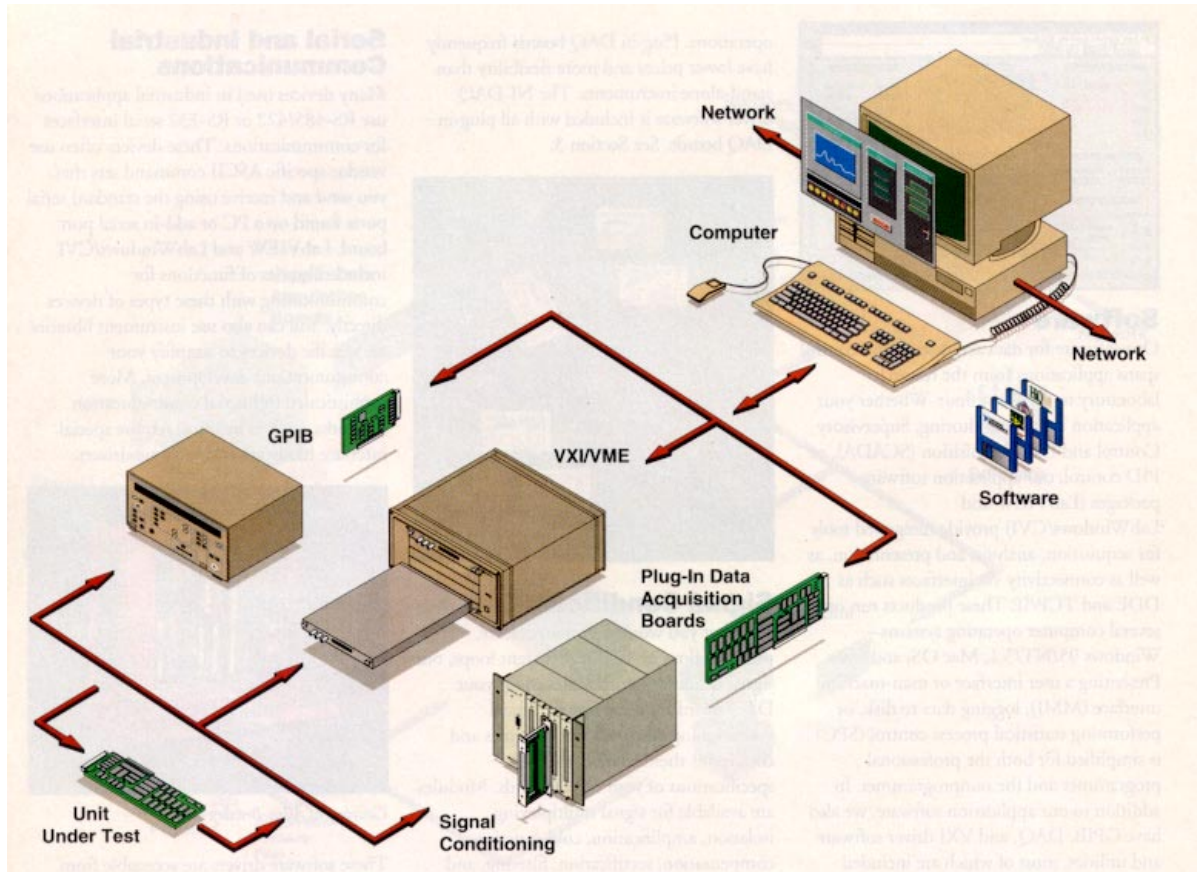


Fig. 1.1 Sistema de Instrumentación virtual (National Instruments Inc.)

Pero no es lo mismo hablar de instrumentación controlable por ordenador que de instrumentación virtual. De la primera a la segunda existe un salto importante, salto que ha sido posible gracias a los avances en el campo de la informática. Hablar de instrumentos virtuales es hablar de un software que se ejecuta sobre el controlador, que permite independizarnos de los instrumentos reales y de la forma de interconectarlos. En muchas ocasiones el usuario final del sistema de instrumentación sólo ve los instrumentos virtuales en la pantalla del ordenador.

La forma habitual de construir un sistema de instrumentación virtual, a partir de los instrumentos controlables que tenemos disponibles y de tarjetas de adquisición conectadas directamente al bus del controlador es utilizar un software comercial que facilite esta tarea. Existen varias posibilidades pero se pueden resumir en dos grandes grupos, los entornos de programación de tipo lingüístico y los entornos de programación gráfica. Además de este software necesitaremos una tarjeta controladora (GPIB en nuestro caso) dentro del ordenador para que actúe de controlador, y los *drivers* de control de los instrumentos que los suele facilitar el fabricante del entorno de programación.

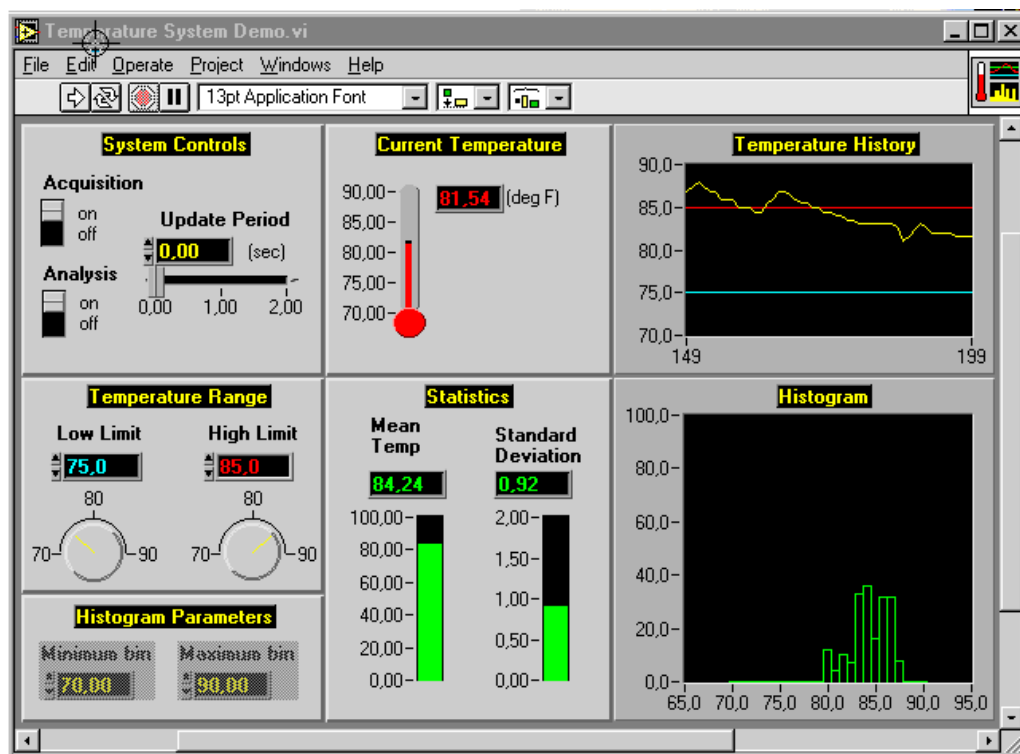
Para la realización de estas prácticas, utilizaremos la tarjeta controladora GPIB- PCIIA de National Instruments, junto con una tarjeta de adquisición de datos del mismo fabricante (PCLab+). El entorno de trabajo es un entorno gráfico de programación, LabVIEW 5.0.

Introducción a LabVIEW 5.0

La palabra LabVIEW esta formada por las iniciales de **L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench. Es un entorno gráfico para el desarrollo de aplicaciones en el campo de la instrumentación, desde la adquisición de datos hasta el control remoto de instrumentos. El entorno dispone de librerías matemáticas para el análisis de datos y de los *drivers* de control de varios instrumentos.

Los programas de LabVIEW se denominan instrumentos virtuales, **VI**, porque la apariencia de su interfase con el usuario es la de un instrumento de laboratorio. Estos VI son equivalentes a las funciones de C o a los procedimientos de Pascal.

Un VI consta de dos partes bien diferenciadas, el **Panel Frontal (Front Panel)** y el **Diagrama de Bloques (Block Diagram)**. El panel frontal es la interfase del programa con el usuario. En él están representadas todas las entradas y salidas del programa. Por analogía a un instrumento real, las entradas del panel frontal se llaman controles y las salidas indicadores. El diagrama de bloques es el código de programación escrito en lenguaje gráfico. Los distintos componentes del diagrama de bloques son los nodos del programa. Los componentes están interconectados unos con otros. Estas interconexiones definen el flujo de datos en el diagrama de bloques. La figura 1.2 contiene el panel frontal y el diagrama de bloques de un VI para el control de temperatura.



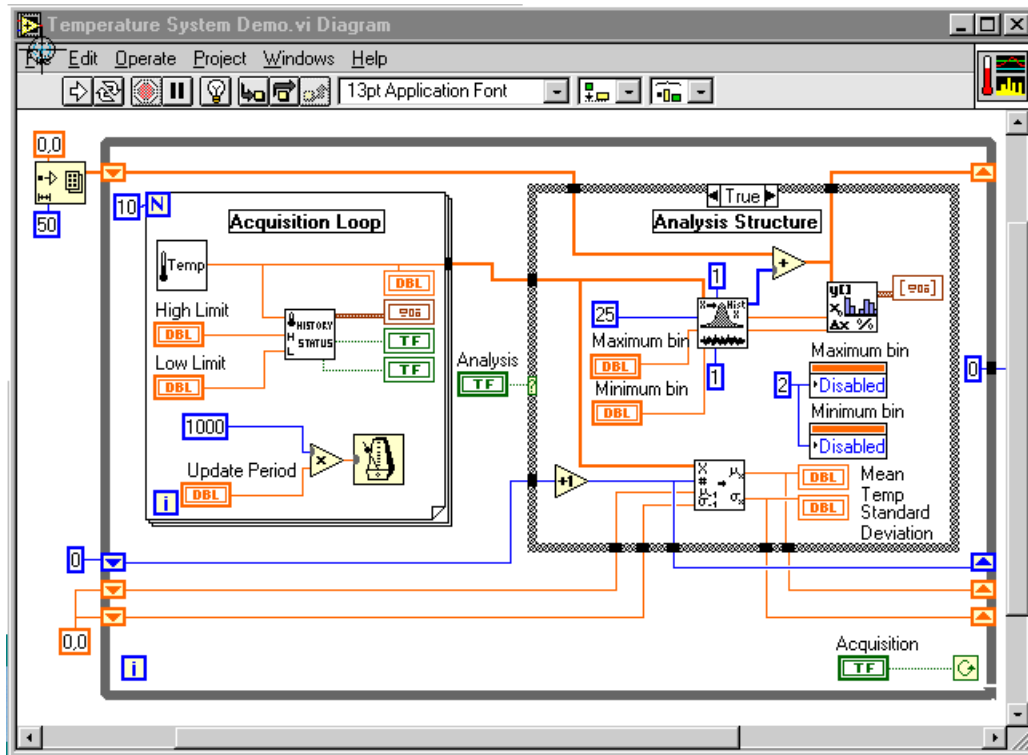


Fig. 1.2 Panel frontal y diagrama de bloques de un VI para el control de temperatura.

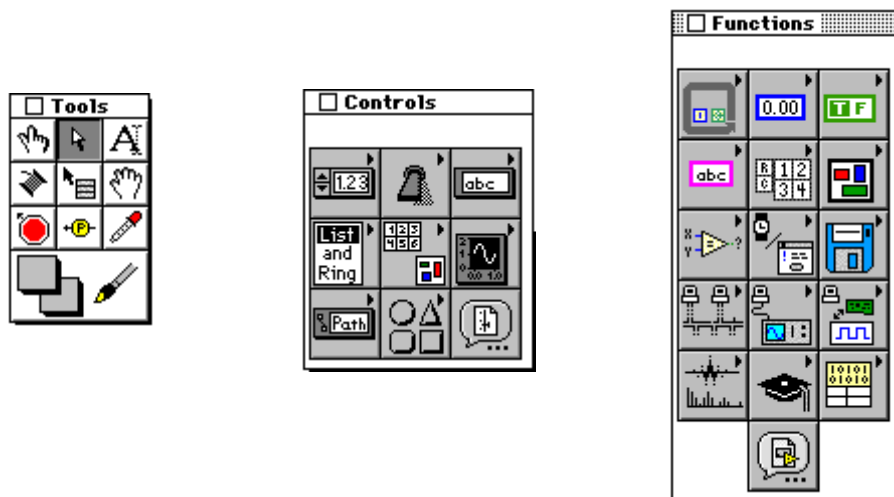


Fig. 1.3 Paletas de herramientas, controles y funciones

Programación con LabVIEW 5.0

Para empezar un programa en LabVIEW debemos seleccionar la opción **New VI** en la ventana que aparece cuando abrimos el programa. Se crearán, entonces, dos ventanas vacías, una correspondiente al panel frontal y una correspondiente al diagrama de bloques.

El diseño del programa se suele empezar en el panel frontal. Se debe decidir cómo será la interfase de usuario, es decir, qué entradas y salidas tendrá el programa. Veamos un ejemplo de cómo se realiza este diseño.

Vamos a diseñar un generador de señal sinusoidal del cual podemos variar la amplitud entre 0 y 10 V. La frecuencia de la señal será fija. Como parámetros de entrada necesitaremos un control para la amplitud y un control de puesta en marcha. Como salida necesitaremos un indicador que nos permita visualizar la señal generada.

Nos situamos sobre el panel frontal. Deben aparecer dos paletas flotantes, una paleta de herramientas que sirve para editar, modificar y depurar VI's y la paleta de controles que sirve para crear el panel frontal (figura 1.3). Si alguna de estas dos paletas no es visible se debe activar **show Tools palette** o **show Controls palette** de la opción **windows** de la barra de menú superior.

En la paleta de controles seleccionaremos los controles numéricos y de entre ellos un control circular (*Knob*) y lo colocaremos sobre el panel frontal. A continuación escribiremos el nombre de este control (amplitud).

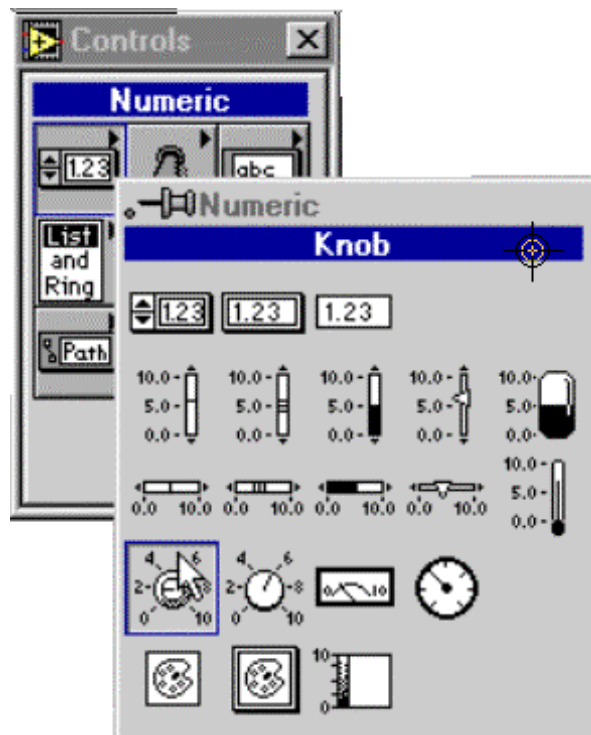


Fig. 1.4. Controles numéricos Podemos observar que al situar un elemento sobre el panel frontal, aparece también en el diagrama de bloques una referencia a este elemento que lleva el mismo nombre. En este caso es un cuadrado con línea doble de color naranja que indica que es una entrada de un número de coma flotante de doble precisión (figura 1.5).

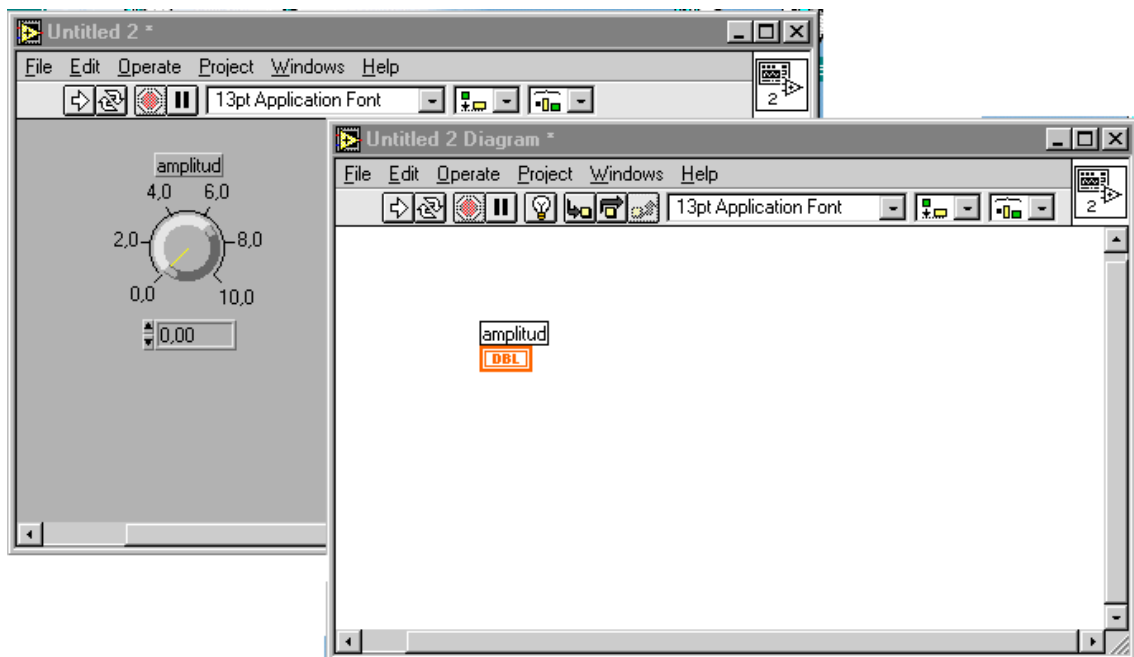


Fig. 1.5. Panel frontal y diagrama de bloques después de insertar el control numérico

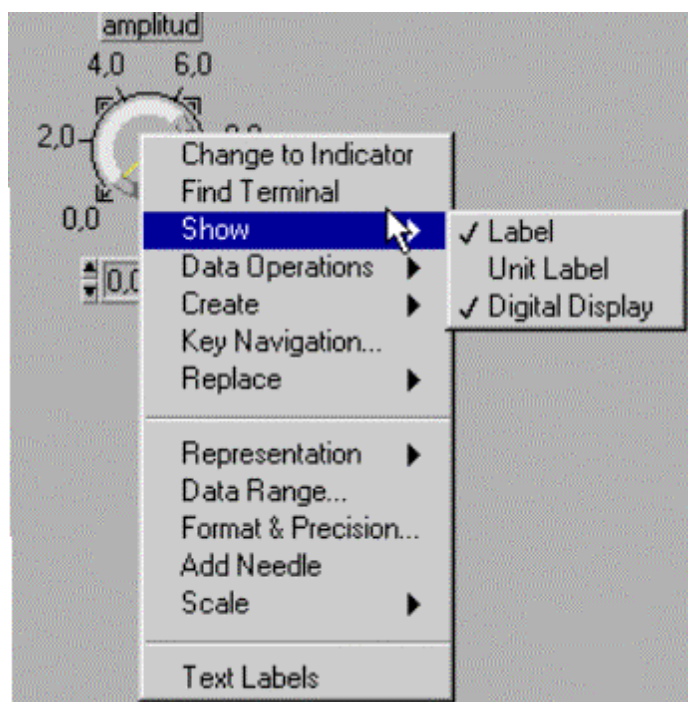


Fig. 1.6 Menú desplegable de un control

Si no se ha podido escribir el nombre del indicador, situarse sobre el mismo y apretar el botón derecho del mouse. Aparecerá un menú desplegable con las opciones de edición de este control (figura 1.6)

Seleccionar Show Label y escribir ahora el nombre del control. Observar que además del control circular existe un indicador digital con el valor de la amplitud seleccionado. Este indicador se puede ocultar si se desea.

Una vez fijado el control de la amplitud de la señal de entrada colocaremos en el panel frontal el control que nos permita la puesta en marcha del generador. En este caso seleccionaremos los controles booleanos de la paleta de controles y escogeremos un botón que nos permita poner en marcha el generador (figura 1.7).

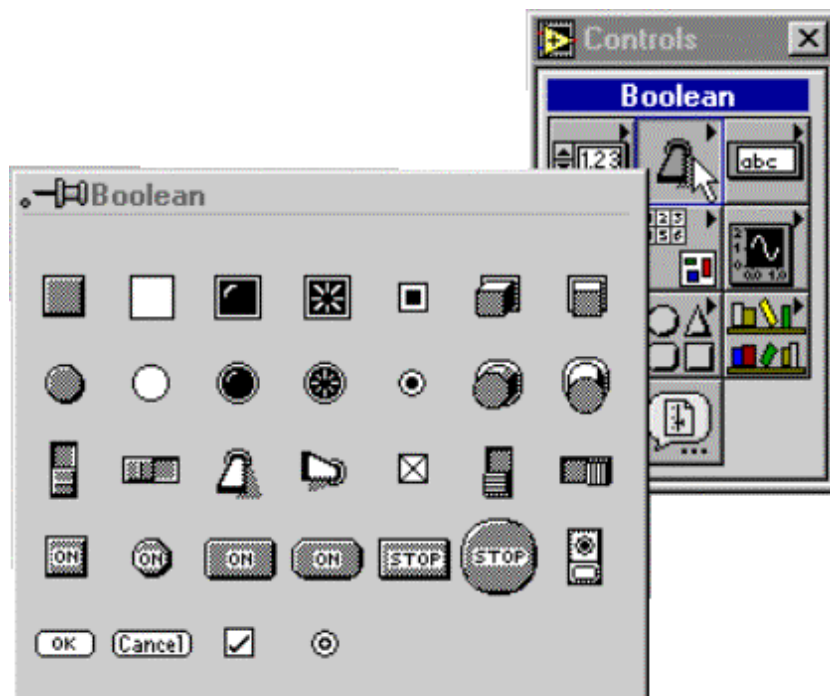


Fig. 1.7 Controles booleanos

Ahora, en el diagrama de bloques ha aparecido una referencia al botón que hemos situado en el panel frontal, un cuadrado con doble línea de color verde con la inscripción T/F (**True/False**) en su interior. La doble línea nos indica que es un control (una entrada) el color verde indica que es un dato booleano (figura 1.8).

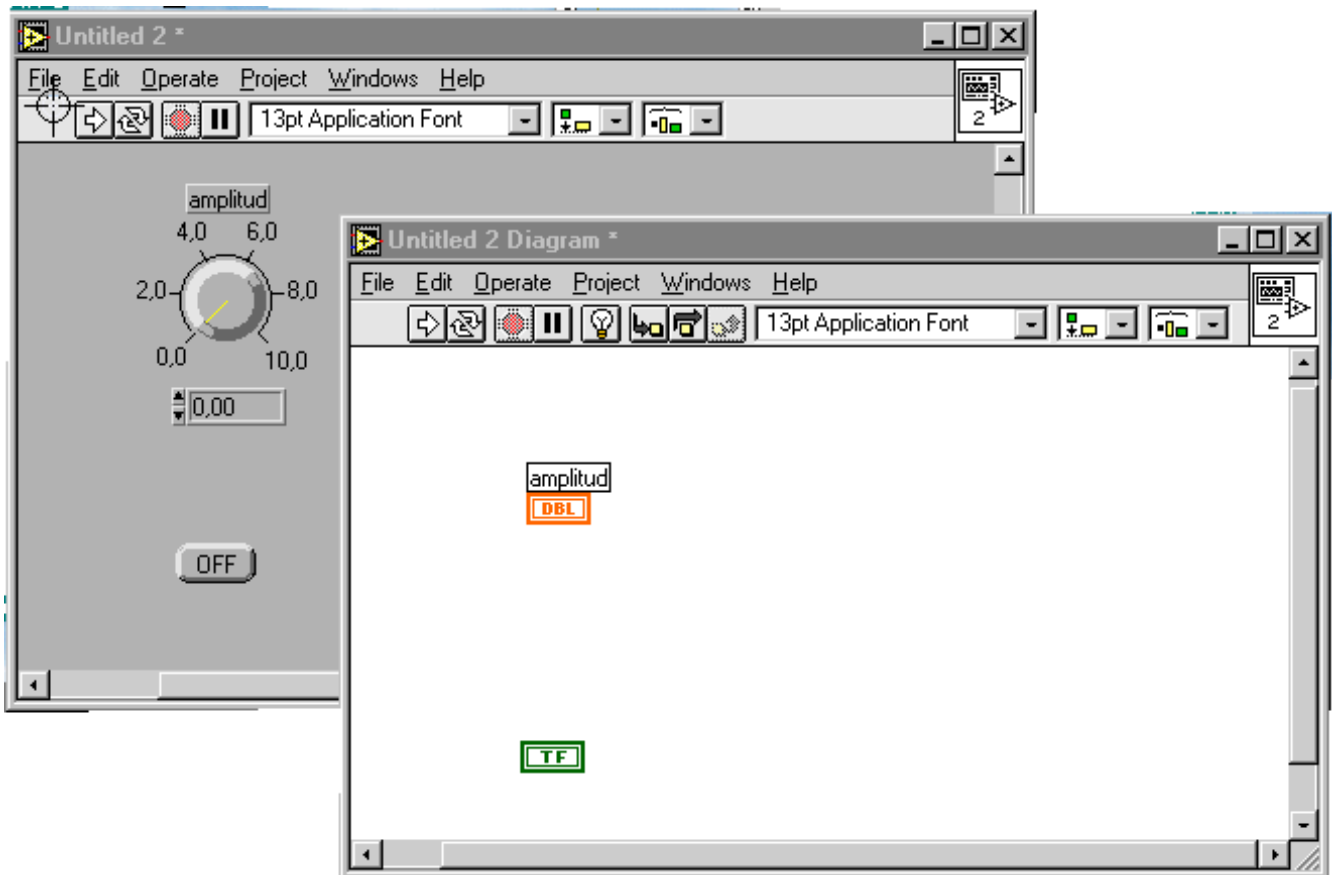


Fig. 1.8 Panel frontal y diagrama de bloques tras insertar los dos controles de entrada

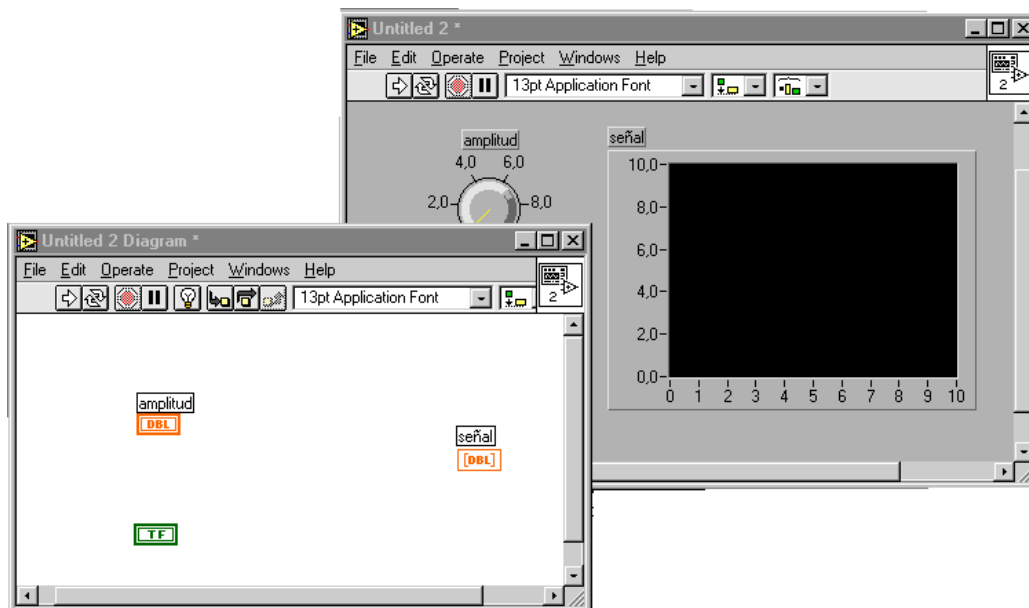


Fig. 1.9 Panel frontal y diagrama de bloques después de insertar los dos controles y el indicador.

Ahora que ya tenemos definidas todas las entradas al programa definiremos la salida. En nuestro caso un indicador que nos permita visualizar la señal generada. En la paleta de controles seleccionaremos el grupo indicadores gráficos y de entre ellos el **waveform graph** (permite dibujar un vector en función de su índice) (figura 1.9).

Una vez más, en el diagrama de bloques ha aparecido una referencia al indicador del panel frontal, en este caso un rectángulo de color naranja con línea simple indicando que es una salida de números enteros.

Ahora que tenemos el panel frontal definido, pasaremos a la construcción del diagrama de bloques. Activaremos la paleta de funciones de la misma manera que la paleta de controles del panel frontal y en ella seleccionaremos la subpaleta de análisis, dentro de ella la generación de funciones y dentro de esta última un VI que nos genera una forma de onda sinusoidal (figura 1.10)

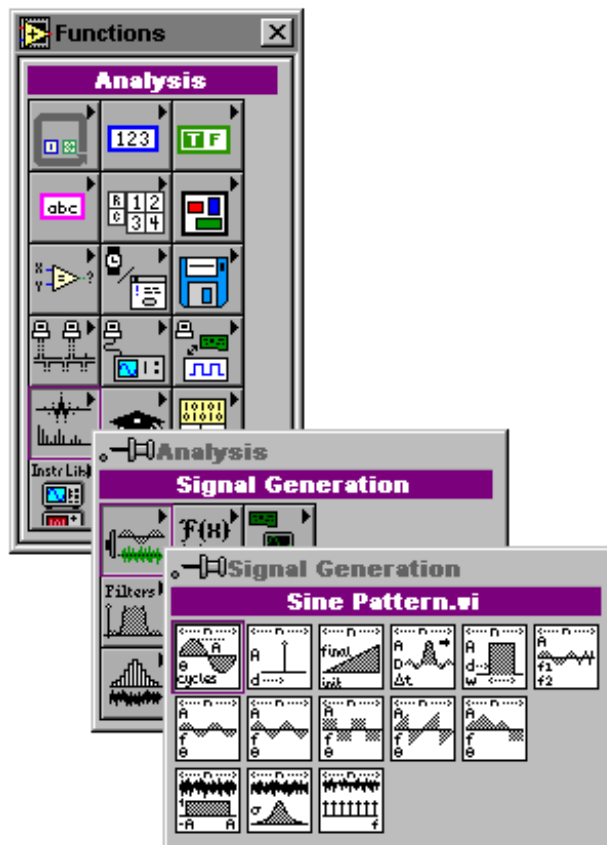


Fig. 1.10 Subpaleta de análisis y subpaleta de generación de funciones.

En el diagrama de bloques, debemos realizar las conexiones de los distintos elementos que indiquen el flujo con que se van a realizar las distintas operaciones. Para establecer estas conexiones, seleccionaremos de la paleta de herramientas el carrito de hilo.

Antes de empezar las conexiones veamos qué entradas y salidas tiene el VI que nos genera la señal senoidal. Para ello nos situamos sobre el icono y apretamos el botón de la derecha del mouse. Aparecerá un menú desplegable donde podemos seleccionar la opción **online help**. A través de ella se activa una ayuda donde aparece una descripción completa de la función.

La figura 1.11 muestra las entradas y las salidas de este VI.

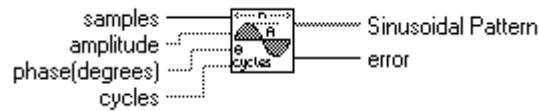


Fig. 1.11 Conexiones del VI que genera la señal senoidal.

Como entradas debemos definir el número de muestras de la señal, la amplitud, la fase en grados y el número de ciclos a visualizar. Como salidas nos da un vector de números correspondientes a las muestras de una señal senoidal y una señal de error.

En nuestro caso, el número de muestras, la fase y el número de ciclos son constantes numéricas. Para fijar su valor, seleccionaremos de la paleta de funciones la opción *numeric* y dentro de ella escogeremos una constante y escribiremos su valor, por ejemplo 1024 muestras, 30° de desfase y 2,5 ciclos. A continuación uniremos cada una de estas constantes con la entrada correspondiente de la función seno mediante el carrito de hilo. Finalmente uniremos la salida de señal con el indicador visual. El aspecto del diagrama de bloques debe ser parecido al de la figura 1.12.

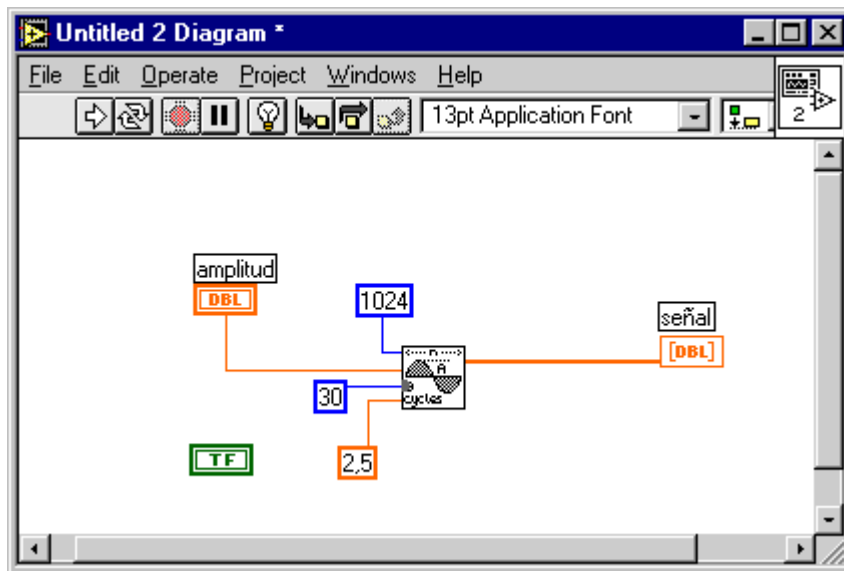


Fig. 1.12 Diagrama de bloques tras realizar las conexiones.

El control del funcionamiento del generador, se realiza a través del interruptor ON/OFF cuya representación en el diagrama de bloques es un indicador booleano (T/F). Este indicador lo utilizaremos como control de un bucle **while** que controle la generación de señal. Para crear este bucle, seleccionaremos la opción estructuras de la paleta de funciones y dentro de ésta el bucle *while*. Nos situaremos en el extremo superior derecho del 'código del programa' y arrastraremos el mouse hasta envolver todo el gráfico que debe controlar el bucle, manteniendo apretado el botón izquierdo del mouse.

Ahora debemos conectar el indicador T/F al control del bucle (una flecha en forma circular de color verde). En la figura 1.13 se encuentra el diagrama de bloques del generador.

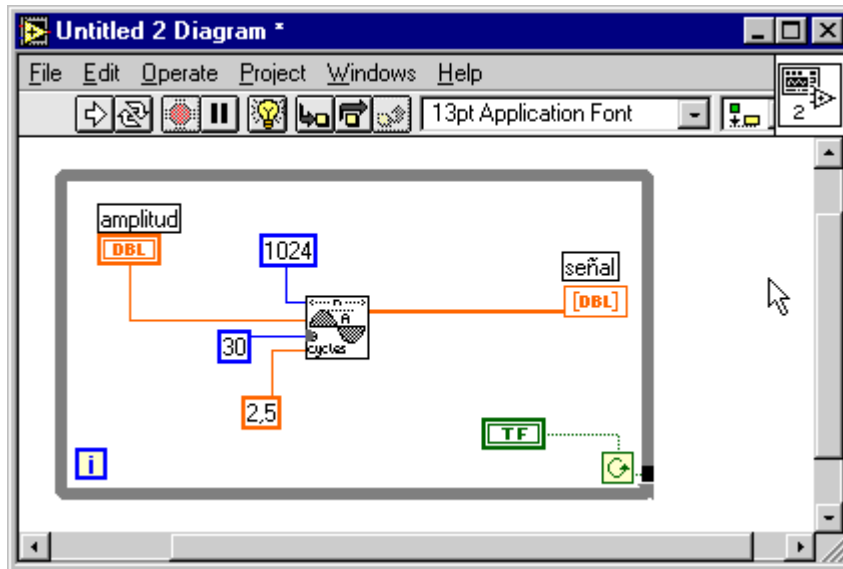


Fig. 1.13 Diagrama de bloques del generador de señal senoidal

Antes de continuar, salvaremos el programa realizado. Para ello vamos al menú y escogemos la opción para guardar el fichero **save as**.




Ejecución de un programa con LabVIEW 5.0

Existen varias formas de ejecutar un programa en LabVIEW. Las distintas opciones pueden encontrarse en la barra de tareas del diagrama de bloques (figura 1.14):










Fig. 1.14 Barra de tareas del diagrama de bloques

A continuación se describen las distintas opciones de ejecución:

-  Ejecución del programa. LabVIEW compilará el programa si es necesario.
-  Mientras se está ejecutando el programa el botón anterior cambia de aspecto y se transforma en la imagen de la izquierda.
- 

Si se está ejecutando un subVI el botón de ejecución se transforma en el de la izquierda.

-  Ejecución continua hasta que se aprieta el botón de pausa o se finaliza la ejecución.
-  Finaliza la ejecución.
-  Botón de pausa.
-  Inicio de ejecución paso a paso. Ejecuta una estructura o un subVI y se para en el siguiente nodo.
-  Ejecuta el primer paso de una estructura o un subVI y se para antes de ejecutar el siguiente paso.
-  Finaliza la ejecución de una estructura, diagrama de bloques o VI y se para.
-  Además de estas opciones se puede ejecutar el programa en modo 'highlight'. Este modo de ejecución permite ver una animación de cómo se ejecuta el VI.

Cuando al ejecutar un VI existen errores aparece una ventana como la de la figura 1.15 con una lista de los errores que se han encontrado.

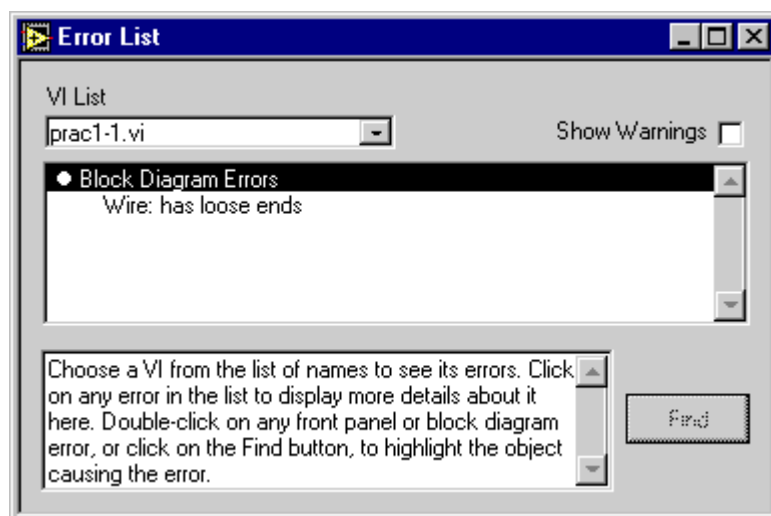


Fig. 1.15 Lista de errores

Si nos situamos con el cursor sobre uno de los errores apretando el botón Find, se indica dónde está el error en el diagrama de bloques.

Aunque un VI no dé errores de ejecución, puede ser que no esté realizando la función que esperábamos. Para poder depurar el funcionamiento existen dos herramientas: **probe** y **breakpoint**. La primera nos permite visualizar el valor de un dato mientras fluye en el diagrama de bloques. La segunda nos permite establecer un punto de interrupción en el programa. Ambas se encuentran en la paleta de herramientas.

En un VI pueden existir varias funciones que se ejecuten simultáneamente. De hecho, el único requisito para que se ejecute un determinado bloque es que tenga los datos disponibles a su entrada. Normalmente el flujo de datos en el VI es de izquierda a derecha; las entradas suelen estar situadas a la izquierda del icono y las salidas a la derecha. Si se desea que se ejecuten ciertos bloques en un orden determinado se debe utilizar una estructura llamada **secuencia**. En la figura 1.16 la secuencia de ejecución de los bloques es A, C, B, D, E porque una entrada de B es la salida de C.

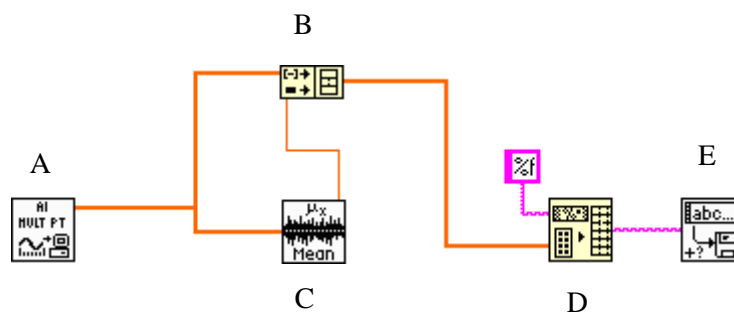


Fig. 1.16 Secuencia de ejecución

Creación de subVI's

Un subVI tiene una función similar a la de las subrutinas en los lenguajes de programación textuales. Cuando un conjunto de operaciones se prevee que se va utilizar en diversas ocasiones, se suelen englobar éstas dentro de un subVI. Así, un subVI es un VI que se llama dentro de otro VI.

Un VI que se usa como un subVI necesita un icono para ser llamado desde el diagrama de bloques. Todos los VI tienen un icono genérico en la parte superior derecha de la pantalla que se puede editar haciendo un doble clic con el botón izquierdo del mouse desde el panel de control.

El editor de iconos permite realizar un dibujo representativo del icono y fijar los terminales de entrada y salida del mismo. Para definir los terminales de entrada y salida del subVI nos situaremos sobre el icono (parte superior derecha de la pantalla) y apretando el botón derecho del mouse escogeremos la opción **show connector**. Aparecerán en lugar del icono la representación de los terminales de entrada y salida del VI. Habrá tantos terminales de entrada y salida como controles e indicadores tenga el VI. El número de terminales de entrada y salida se puede modificar colocándonos sobre la representación de los terminales de entrada y salida, apretando el botón derecho del mouse y escogiendo la opción **patterns**.

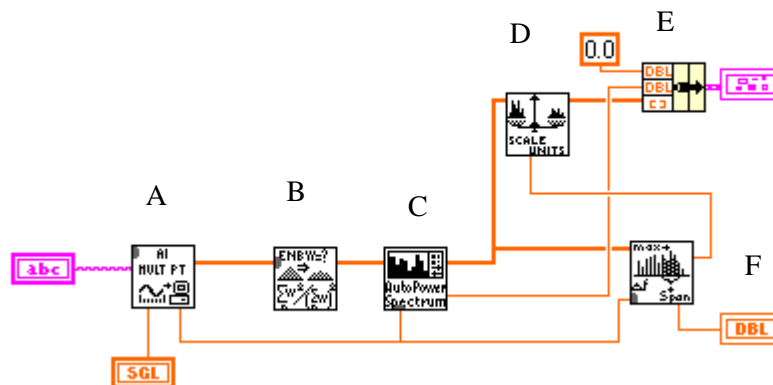
Cada control se debe asignar a un terminal de entrada y cada indicador a un terminal de salida. Esta asignación se hace mediante la herramienta **wire** (carrete de hilo) que aparece por defecto en la pantalla cuando estamos en la opción **show connector**. Para realizar la asignación nos situaremos primero sobre un terminal y lo seleccionaremos, después nos colocamos sobre un control o un indicador y lo seleccionaremos también. El control o indicador y el terminal quedarán unidos por una línea. Esta operación se debe repetir hasta que todos los indicadores, controles y terminales estén interconectados.

Descripción del entorno informático de trabajo del laboratorio

Los puestos de trabajo disponen de un ordenador basado en un microprocesador Intel Pentium. El sistema operativo es Windows95 y todos los puestos de trabajo están conectados mediante una red telemática a la UPCNet. Cada usuario dispone de un directorio de trabajo en este servidor, de forma que es accesible desde cualquier puesto de trabajo o desde el exterior del laboratorio. Se recomienda hacer copias de seguridad en diskettes.

Trabajo de laboratorio

1. Realizar en el cuaderno de laboratorio un gráfico descriptivo de los componentes del sistema de instrumentación virtual del puesto de trabajo (controlador, instrumentos, conexiones)
2. Iniciar el entorno de trabajo LabVIEW y explorar las diferentes utilidades en las paletas de herramientas, controles y funciones. ¿Qué son los bloques de color amarillo que aparecen en algunas subpaletas de la paleta de funciones? (busque la respuesta con la ayuda del programa)
3. Construir el generador de señal sinusoidal tal como se ha indicado en los apartados anteriores. Probar los distintos modos de ejecución. Insertar algún probe y algún breakpoint para ver cómo funcionan. Ejecutar el programa en modo highlight y observar el flujo de datos.
4. Decidir cuál es el orden de ejecución en el siguiente programa:



5. Diseñar un generador virtual de señales con las siguientes características:

Señales: sinusoidal, cuadrada y triangular
 Amplitud: variable entre 0 y 10 voltios
 Frecuencia: variable entre 0 Hz y 10 kHz

La presentación de la señal se realizará en el dominio temporal. Para la realización de este generador partiremos del generador de señal sinusoidal diseñado en los apartados anteriores.

5.1 Determinar qué controles e indicadores tendrá el generador de señal virtual y su módulo de presentación.

5.2 Realizar un esbozo del diagrama de bloques del generador donde se pueda apreciar cuál será el flujo de datos.

5.3 Modificar adecuadamente el panel frontal para que sea posible elegir entre las señales sinusoidal, triangular y cuadrada.. (Utilizar el **help on line** para conocer el funcionamiento de los distintos tipos de selectores, **'list and ring'** para el panel frontal).

5.4 Modificar el diagrama de bloques para que sea posible generar las tres señales. (Cambiar el VI de la función seno por el VI sine wave) y añadir los VI correspondientes a la señal cuadrada y triangular. Se recomienda utilizar la estructura de datos **case**.

5.5 Modificar el panel frontal y el diagrama de bloques para que sea posible seleccionar la frecuencia de la señal (prestar atención a cómo está definida la frecuencia de la señal en los VI que generan las señales).

5.6 Modificar el panel frontal y el diagrama de bloques para que el eje vertical de la presentación de la señal temporal esté entre -10 V y 10 V y el eje horizontal en unidades de tiempo. Para ello lo más recomendable es utilizar una estructura cluster que contiene varios tipos de datos distintos y utilizar como herramienta de presentación un **XYgraph**, que permite realizar gráficos cartesianos con dos ejes.

6. Crear un SubVI con toda la parte correspondiente a la generación de señal. Sustituir la parte de generación de señal por el SubVI creado.

Trabajo opcional

1. Modificar el panel frontal y el diagrama de bloques para obtener el espectro de la señal y su presentación. Actuar sobre los ejes para que las unidades de la presentación sean las pertinentes.

2. Diseñar un VI que sea un generador de ruido. Considerar ruido uniforme y que la amplitud de ruido pueda variar entre 0 y 100% de la amplitud de la señal.

3. Crear un subVI a partir del VI anterior e incorporarlo al generador de funciones.

Práctica 2

Introducción a IEEE-488.2 y VISA. Identificación automática de recursos en sistemas de instrumentación

Objetivos

En esta sesión de laboratorio nos vamos a centrar en el uso de los procedimientos que la norma IEEE-488.2:1992 especifica para realizar la identificación e inicialización de sistemas de instrumentación. Como el sistema de instrumentación usado en los puestos de trabajo del laboratorio será el mismo durante todo el curso, disponer de un método para detectar si los instrumentos están presentes puede ser de gran ayuda si el *vi* desarrollado se utiliza como primer paso en todos los desarrollos que se hagan en el futuro.

También se introducirá la especificación VISA (Virtual Instrument Standard Architecture) que pretende ser un método independiente del software, la plataforma y el tipo de BUS de instrumentación para el control de instrumentos programables.

Introducción a IEEE-488.2

La norma IEEE-488.2:1992 define la sintaxis de las órdenes para realizar la programación de los instrumentos conectados a un bus GPIB (IEEE-488.1:1987), un conjunto de órdenes comunes (*common commands*) que hacen referencia a la identificación, estado, etc. de los instrumentos, y un conjunto de procedimientos de bus que los controladores compatibles con la norma deben poder realizar.

Cada uno de estos procedimientos se ha realizado, en labVIEW, como un VI. En otros lenguajes y entornos de programación estos procedimientos se realizan como funciones.

En la figura 2.1 se pueden ver el conjunto de VI disponibles en LabVIEW[®]5.0, en la librería INSTRUMENT I/O que realizan los procedimientos definidos en IEEE-488.2

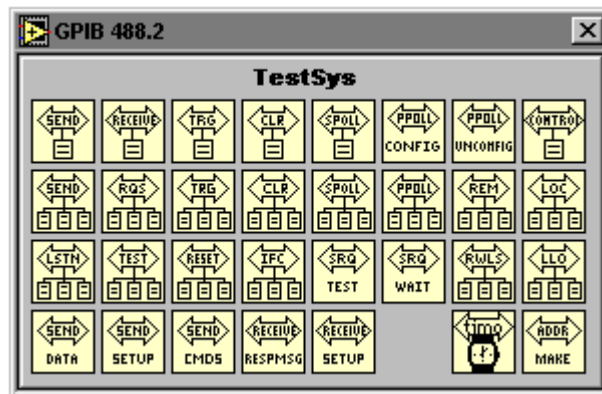


Fig. 2.1 Funciones de LabVIEW 5.0 que realizan los procedimientos descritos en IEEE-488.2

Estos VI se pueden clasificar, básicamente, en 3 grupos:

1. Funciones de comunicación: sirven para enviar comandos y recibir datos de los instrumentos, p. ej. SEND, RECEIVE.
2. Funciones de configuración del sistema: permiten identificar los instrumentos presentes, inicializar el sistema, inicializar dispositivos, etc., p.ej. FINDLSTN, RESETSYS, SEND IFC.
3. Funciones de gestión de interrupciones: permiten detectar y gestionar las peticiones de servicio (Service Request, SRQ) que efectúan los instrumentos, p.ej. SERIALPOLL, TESTSRQ.

En esta sesión utilizaremos las funciones FINDLSTN, SEND y RECEIVE. La función FINDLSTN se utiliza para determinar qué direcciones de las 31 posibles en un bus GPIB están ocupadas por un instrumento, es decir, para determinar qué instrumentos están presentes, conectados al bus y funcionando. Para ello, el controlador del BUS envía un LAG (*Listen Address Group*) a cada una de las posibles direcciones. Si en alguna de ellas hay un instrumento funcionando, éste, al cambiar de estado IDLE a LADS (*Listener Addressed State*) cambia el valor de las líneas NRFD y NDAC. El controlador monitoriza el estado de estas líneas y decide si hay un instrumento presente o no.

El VI de LabVIEW que realiza esta función puede verse en la figura 2.2. Las entradas necesarias son un vector con la lista de direcciones donde buscar instrumentos, el número máximo de instrumentos que puede encontrar y el número de la tarjeta controladora a utilizar. Como salidas nos ofrece un vector con la lista de direcciones en la que hay un instrumento presente, el número de instrumentos encontrados y el estado del bus GPIB después de realizar el comando.



Fig. 2.2 Forma del VI que realiza la función FindLstn en LabVIEW

Además de estas entradas, podemos ver que éste VI dispone de una entrada y una salida de error. Estas entradas/salidas pueden dejarse sin conectar pero es recomendable no hacerlo por varios motivos. En primer lugar ofrecen un mecanismo de secuenciación de las acciones ya que en LabVIEW un VI no se ejecuta hasta que todas las entradas están disponibles. En segundo lugar, los VI están diseñados para dejar pasar el código de error (si es distinto de 0) de forma transparente. De esta forma, en un caso simple, podemos colocar un monitor de errores al final de la cadena de órdenes para observar cualquier error que se haya producido en algún punto de ésta.

La función anterior nos proporcionará un vector con las direcciones en las que se ha detectado algún instrumento. Es posible que instrumentos antiguos, no compatibles con IEEE-488.2, no se comporten de la forma esperada. Una vez obtenida la lista con las direcciones puede ser de utilidad determinar qué instrumento hay en cada dirección. IEEE-488.2 define una orden multibyte básica obligatoria de identificación (***IDN?**) a la que los instrumentos deben responder con una cadena de caracteres que los identifique. Esta cadena debe contener 4 campos, separados por comas:

[Fabricante],[Modelo],[Nº serie | 0],[ver. de firmware | 0]

el contenido de los campos puede ser cualquier carácter alfanumérico excepto la coma o el punto y coma.

Para enviar la orden ***IDN?** a los instrumentos previamente detectados con FINDLSTN usaremos la función genérica de la librería 488.2 para enviar cadenas de caracteres a un dispositivo en el bus. Esta función puede verse en la figura 2.3.

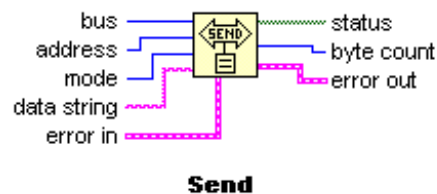


Fig. 2.3 Función para enviar una cadena de caracteres a un instrumento conectado a un bus IEEE-488

Debemos especificar la tarjeta de control usada (0 por defecto), la dirección del instrumento, la cadena a enviar y el modo del terminador. Se recomienda acabar las transmisiones con un carácter **<LF>**. Nos devuelve el estado del BUS y el número de bytes transmitidos. Al igual que FINDLSTN, dispone de una entrada y una salida de error.

La respuesta de los instrumentos la recogeremos con la función RECEIVE, de la misma librería, que tiene un formato análogo. Debe tenerse en cuenta que, por regla general, las funciones que esperan leer información del bus disponen de una variable de control que especifica el número máximo de caracteres a leer. Esta variable debe inicializarse necesariamente a un valor superior al número de caracteres esperado. El valor por defecto es cero, de forma que no se leería ningún dato a través del bus.

Introducción a VISA

La especificación VISA (*Virtual Instruments Standard Architecture*) fue introducida a mediados de la década de los 90 con dos objetivos fundamentales: independizar la estructura de

los VI o *drivers* de instrumentos del tipo de bus o conexión física utilizada y aumentar la portabilidad de las aplicaciones entre plataformas distintas. El primero de los objetivos se enmarca claramente dentro de las aplicaciones que usan el bus VXI. Un mainframe VXI puede estar conectado al controlador de formas diversas: GPIB, MXI, etc, o incluso puede el controlador estar integrado en el sistema. Esto implica, sin VISA, usar programas distintos, para un mismo instrumento, dependiendo del modo de conexión.

La portabilidad se consigue definiendo un conjunto reducido de operaciones (o funciones) soportadas por VISA y que deben estar implementadas en cualquier plataforma y entorno compatible. La realización de las funciones VISA en LabVIEW puede verse en la figura 2.4. En esencia existen funciones para abrir una sesión VISA, cerrarla, enviar y recibir datos de un instrumento, leer el estado de un instrumento e iniciar una medida. Además existen una serie de funciones para el tratamiento de eventos (*Event Handling*) que permiten definir mecanismos de tratamiento de interrupciones o peticiones de servicio de los instrumentos.

Este conjunto reducido de funciones debe permitir el control de cualquier instrumento cuyo mecanismo de comunicación esté basado en mensajes (todos los IEEE-488, RS-232 y algunos VXI). Para incluir los instrumentos VXI basados en registros se han definido dos subgrupos adicionales de funciones, lo que incrementa considerablemente el número inicial, reducido. También se han añadido funciones para controlar las características de los puertos RS-232.

El corazón de VISA es el gestor de recursos (*Resource Manager*). Las llamadas a las funciones de la librería no son más que llamadas a este gestor de recursos, que realiza el trabajo. Las funciones, de hecho, son objetos (desde un punto de vista informático) con propiedades, estructuras de datos y operaciones definidas en el contexto de gestor de recursos. En la terminología VISA se entiende por recurso un instrumento conectado a alguno de los buses disponibles. Actualmente se soportan IEEE-488, VXI y RS232. Los ejemplos y controladores de instrumento que acompañan este libro necesitan la versión 2.0 de la especificación VISA de NI.

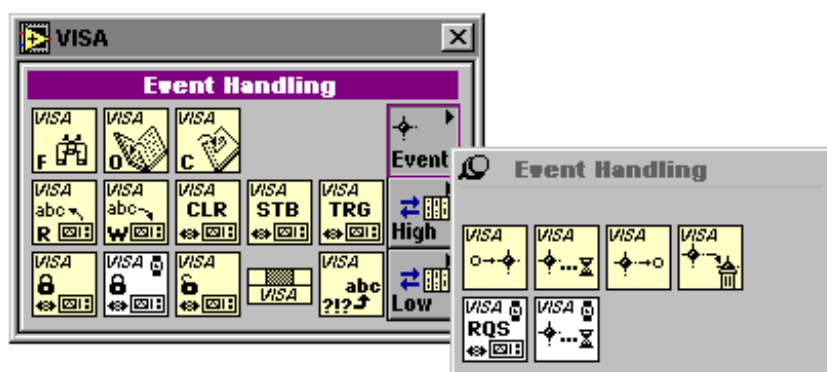


Fig. 2.4 Conjunto de funciones VISA básicas y para el tratamiento de eventos (interrupciones) en LabVIEW

De manera similar al procedimiento FINDLSTN de IEEE-488.2, también existe en VISA una función encargada de determinar qué recursos hay conectados al sistema. En este caso, no obstante, la lista de recursos no se limitará a los conectados al bus IEEE-488 sino que podrá contener todos los presentes en los distintos buses.

Puede verse la forma del VI de LabVIEW que realiza esta función en la figura 2.5. Debemos suministrar una cadena de caracteres a la función para indicarle qué tipo de recursos debe buscar. Por ejemplo `?*INSTR` significa buscar todos los instrumentos disponibles y `GPIB[0-9]*::?*INSTR` significa buscar sólo los instrumentos conectados una de las tarjetas controladoras GPIB, entre 0 y 9.



Fig. 2.5 Función para determinar los recursos conectados al sistema

La función nos devuelve el número de recursos encontrados y la especificación de los mismos, en formato

TIPO-DE-INTERFASE[Nº TARJETA]::DIRECCION::CLASE

Así por ejemplo `GPIB0::7::INSTR` indicaría un instrumento que se encuentra en la dirección 7, conectado a la placa controladora GPIB nº 0.

Una vez determinados los recursos existentes, para poder comunicarnos con un instrumento debemos abrir una sesión (VISA OPEN) con el gestor de recursos. Una vez finalizada la comunicación con el instrumento cerraremos la sesión (VISA CLOSE).

El diagrama de la figura 2.6 nos muestra el procedimiento para abrir una sesión, enviar datos al instrumento, recibir la respuesta y finalmente cerrar la sesión. La función de apertura (VISA OPEN) nos devuelve un identificador de sesión, que debemos suministrar a todas las funciones posteriores que deban comunicarse con el recurso abierto. Además, la función VISA OPEN requiere una entrada que le especifique la clase de instrumento que se va a abrir. La clase, de hecho, está incorporada al descriptor del recurso, como se ha visto anteriormente, pero esto no es obligatorio en la especificación actual de VISA, aunque se espera que lo sea en el futuro. El objeto que debemos colocar en esta entrada de VISA OPEN lo podemos encontrar en la paleta de CONTROLS, bajo PATH & REFNUMS. Por defecto la clase suministrada es `INSTR`, pero podemos editar el control para especificar otro valor.

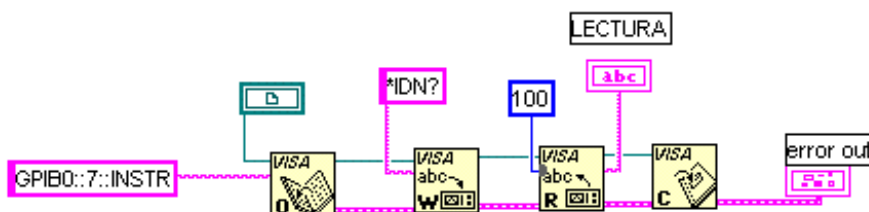


Fig. 2.6 Ejemplo de diagrama que muestra como abrir una sesión con un instrumento GPIB, como enviarle datos, leer su respuesta y cerrar la sesión, usando los VI de la librería VISA

Identificación automática de recursos

En un sistema real de instrumentación es importante identificar los instrumentos disponibles antes de intentar realizar las tareas de medida. Un instrumento que no funciona, está apagado, desaparecido o cuya dirección ha sido cambiada por alguien, puede ser una fuente importante de errores de difícil detección si no se han previsto mecanismos específicos para ello. La forma más simple es determinar, antes de empezar a medir, si todos los instrumentos responden y tienen asignadas las direcciones esperadas.

Una vez realizada la comprobación anterior, si todo es correcto, se procederá a realizar las medidas. Si la información sobre la configuración del sistema no es la esperada se puede proceder de varias formas, que dependen de la capacidad de los instrumentos y buses utilizados y de la pericia del programador.

Algunos instrumentos IEEE-488 permiten que su dirección sea cambiada por el controlador, de forma dinámica, y IEEE-488.2 especifica los procedimientos para realizarlo, aunque esta capacidad no es habitual. Si se detecta que todos los instrumentos están presentes pero las direcciones no son las esperadas, se puede proseguir con las medidas previstas, teniendo en cuenta este cambio. Esta manera de proceder condiciona fuertemente el algoritmo de medida. La forma más simple de proceder es dar una señal de error y mostrar al usuario la información recogida y la esperada, de forma que sea el operador humano quien solviente la situación.

Trabajo previo

Para cada uno de los instrumentos del puesto de trabajo determinar, a partir de la información contenida en los manuales: la dirección GPIB de defecto, cómo puede cambiarse y la compatibilidad con IEEE-488.2

Trabajo en el laboratorio

Identificación mediante el procedimiento FindLstn

Deberá realizarse un VI que determine en qué direcciones de las 31 posibles en un bus GPIB hay un instrumento conectado, usando la función FINDLSTN de la librería IEEE-488.2. Debe tenerse en cuenta que la placa controladora tiene la dirección 0. Si se intenta determinar si hay un instrumento presente en esta dirección el bus puede colapsarse. También se debe tener en cuenta que la dirección 31 no está disponible para asignarla a un instrumento, sino que se utiliza para enviar un mensaje de UNLISTEN o UNTALK a todos los instrumentos.

Una vez obtenido el vector con las direcciones de los instrumentos presentes, deberán identificarse estos. Lo más simple es enviar la orden ***IDN?** y obtener la respuesta del instrumento. Si alguno de los instrumentos no es compatible IEEE-488.2 la respuesta a esta orden puede no ser la esperable. Este tipo de casos deben tratarse manualmente.

Identificación usando las funciones VISA

Se pretende realizar lo mismo que en el apartado anterior, pero usando exclusivamente las funciones de la librería VISA.

Trabajo optativo

Generación de alarmas

Usando uno de los procedimientos anteriores, una vez obtenida la identificación de los instrumentos, comparar la asignación de direcciones con las de defecto. Si no coinciden se deberá generar una alarma acústica (librería de ADVANCED FUNCTIONS) y presentar en pantalla la información necesaria para que el usuario pueda corregir la situación.

Crear un SubVI con una salida booleana que contenga el diagrama anterior. Este subVI se podrá usar en las prácticas futuras al inicio de la sesión, para comprobar que el sistema funciona correctamente.

Práctica 3

Control de instrumentos a bajo nivel. Peticiones de servicio.

Objetivos

Con las herramientas actuales, la mayor parte de las aplicaciones se desarrollarán usando VI de alto nivel (*drivers* de instrumento o aplicaciones de más alto nivel). A pesar de ello, no es infrecuente tener que acceder a funciones de un equipo que no están contempladas en estas herramientas de alto nivel, o tener que depurar fallos en ellas, o incluso tener que desarrollar las propias. Uno de los objetivos de esta práctica es familiarizarse con las herramientas de bajo nivel disponibles en LabVIEW.

El segundo objetivo es aprender a utilizar una de las potencialidades mayores del BUS IEEE-488 (y también de otros buses): la capacidad de generar interrupciones para notificar eventos al controlador. Esta potencialidad se suele infrautilizar, muchas veces por un cierto halo de misterio que la envuelve y lo críptico de las funciones del controlador que permiten tratar estos eventos.

Control de instrumentos a bajo nivel

El concepto de control a *bajo nivel* es confuso y cambiante con la tecnología, los programas de aplicación disponibles y el tipo de usuario. En el contexto de la programación en LabVIEW entendemos por bajo nivel la programación de las funciones de medida y la recogida de datos de los instrumentos sin utilizar los *drivers* específicos de estos. Para realizar esta comunicación usaremos las funciones genéricas para enviar cadenas de caracteres a un instrumento y las funciones de lectura.

Existen diversas familias de funciones en LabVIEW que realizan esta operación. En la practica 2 se han visto dos grupos de ellas: las basadas en GPIB 488.2 y las basadas en VISA. Existen además funciones (antiguas) basadas en GPIB, anteriores a la aparición de la norma IEEE-488.2 y cuyo uso no se recomienda, excepto por razones de compatibilidad con aplicaciones antiguas.

Sintaxis de las órdenes de medida. SCPI

Para poder programar una función de medida en un instrumento debemos elegir una función que nos permita enviarle una orden y debemos saber qué orden activa cada posible medida que el instrumento es capaz de realizar. Desde los inicios de los instrumentos basados en GPIB ha habido un cierto desorden en cuanto a la sintaxis y al contenido de las órdenes. Instrumentos de funcionalidad similar de distinto fabricante, o incluso del mismo fabricante, usaban conjuntos de órdenes totalmente distintos, lo cual suponía un gran quebradero de cabeza para el programador. La norma IEEE-488.2-1987 primero y el consorcio SCPI (Standard Commands for Programmable Instrumentation) después racionalizan un poco este panorama.

IEEE-488.2 define la **sintaxis** que deberán usar los comandos de programación, pero sin definir su contenido. SCPI adopta la sintaxis de IEEE-488 y define los contenidos a través de una serie de modelos genéricos de instrumento.

De forma muy sucinta, la sintaxis definida en IEEE-488.2 puede explicarse de la siguiente forma (ver figura 3.1):

Un mensaje está constituido por un mensaje de programación (Program Message: PM) y un terminador. Un mensaje de programación está constituido por un conjunto de unidades de programación (Program Message Units: PMU), separadas por un punto y coma. Un terminador puede ser la activación de la línea EOI, un carácter <LF> o ambos simultáneamente.

Una unidad de programación consta de una cabecera (Header) terminada con un carácter de interrogación, si esperamos respuesta y, opcionalmente, unos datos, separados de la cabecera por un espacio en blanco. Los datos pueden ser numéricos o alfanuméricos y van separados por comas.

Una cabecera es un conjunto de nemotécnicos separados por un carácter ":". Opcionalmente un mnemotécnico puede llevar un carácter "*" al inicio, para indicar que es una orden básica definida en la norma IEEE-488.2.

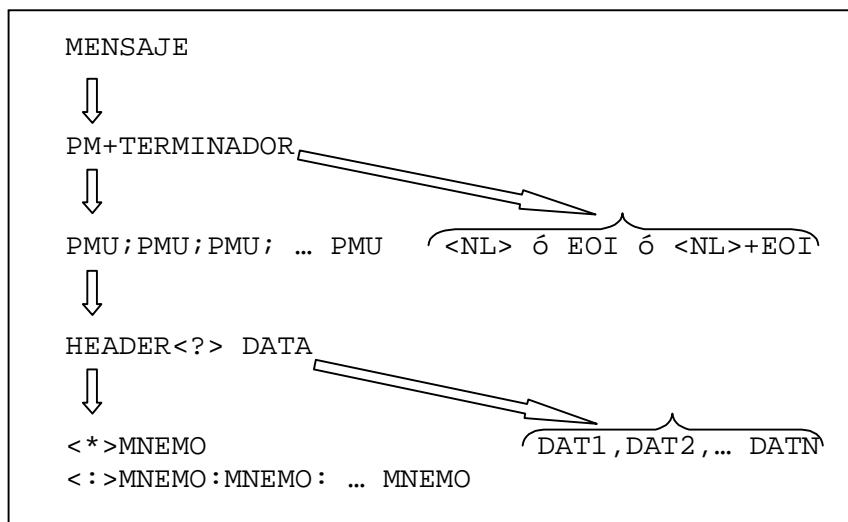


Fig. 3.1 Representación esquemática y simplificada de la sintaxis definida en la norma IEEE-488.2-1987

Así por ejemplo, la orden ***IDN?** usada en la práctica anterior se interpreta dentro de este esquema como: un PM que contiene una única PMU. No hay datos. El interrogante significa que esperamos respuesta del instrumento y el asterisco que se trata de una orden básica obligatoria definida en la norma.

Si quisiésemos configurar un instrumento hipotético para medir resistencia a 4 hilos, en la escala de defecto y con una resolución de 1 m Ω , escribiríamos:

CONFIG:RES:4HIL DEF,0.001

Si además queremos que el instrumento realice la medida y nos devuelva el resultado, la hipotética orden sería:

MEDIDA:RES:4HIL? DEF,0.001

Si lo que queremos es configurar el instrumento y realizar 3 medidas consecutivas, la cadena podría ser:

CONFIG:RES:4HIL DEF,0.001;MEDIDA?;MEDIDA?;MEDIDA?

En este último caso hemos realizado las peticiones de medida dentro del mismo mensaje, con varias PMU's. Podríamos hacer lo mismo enviando mensajes distintos, con una única PMU en cada uno y realizar la lectura después de cada petición.

Éste es el estado de las cosas después de IEEE-488.2, es decir, los nemotécnicos que programan una función de medida específica son criterio del fabricante del instrumento. La "norma" SCPI fue adoptada por un consorcio de fabricantes de instrumentos para definir el contenido de los mnemotécnicos. La filosofía de SCPI se basa en unos modelos de instrumentos, como el de la figura 3.2, de forma que instrumentos equivalentes a algún nivel del modelo deben aceptar las mismas órdenes para realizar la misma medida.

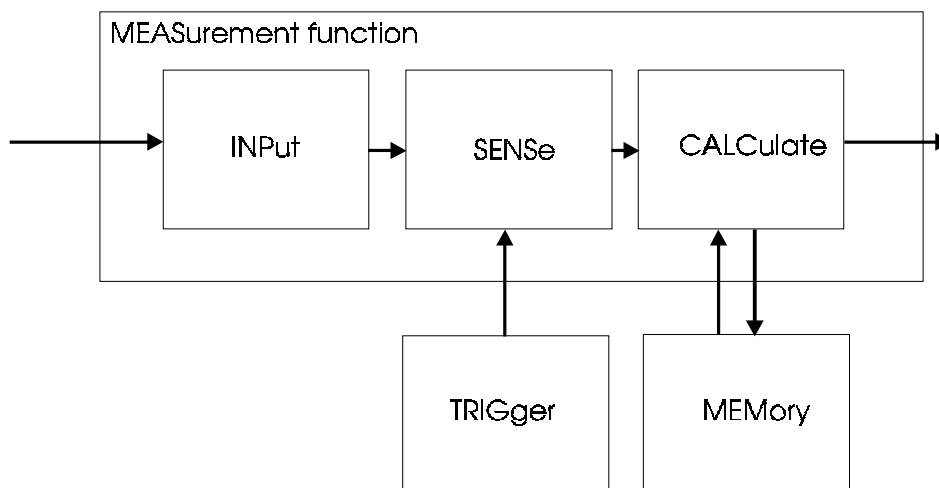


Fig. 3.2 Función de medida con sus 3 componentes (entrada, conversión y cálculo), según el modelo SCPI y la interrelación con los subsistemas de disparo y memoria.

Así por ejemplo un multímetro capaz de medir tensiones continuas y un osciloscopio capaz de realizar la misma medida deberían aceptar la orden

MEAS:VOLT:DC? 10,0.1

referida al subsistema de medida. En cambio, si bajamos de nivel, en el subsistema de *SENSE*, el osciloscopio no tiene porqué entender la orden

SENS:VOLT:DC:NPLC 10

en la que se programa el tiempo de integración del convertor A/D del multímetro, porque el osciloscopio usa un procedimiento de medida distinto.

Los documentos de SCPI especifican todas las posibles órdenes que deben aceptar cada uno de los subsistemas del modelo de instrumento genérico. Un instrumento real aceptará sólo un subconjunto de estas órdenes.

Envío de la información

Para enviar estos comandos a los instrumentos usaremos las funciones de la librería GPIB-488.2 o bien de la librería VISA. En la figura 3.3 podemos ver las formas que toman en LabVIEW estas funciones.

Aunque parezcan equivalentes, estas funciones no son intercambiables en todos los casos. La función WRITE de VISA asume que estamos enviando información a un dispositivo que se comunica por mensajes, esto es, compatible con la sintaxis de IEEE-488.2. Aunque parezca contradictorio, la función SEND de la librería GPIB 488.2 no hace esta asunción.

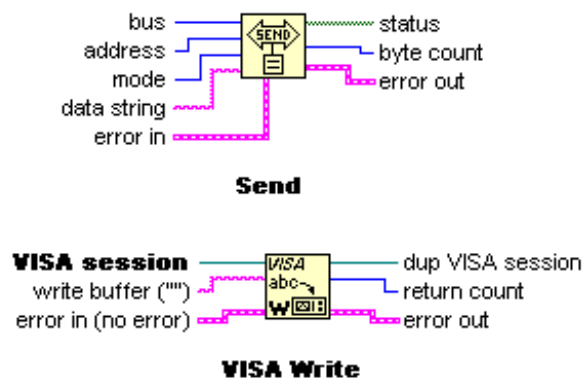


Fig. 3.3 Funciones para enviar datos a un instrumento, de las librerías GPIB 488.2 y VISA.

La diferencia estriba en que en la función SEND podemos especificar el terminador del mensaje mediante la entrada MODE (0: sin terminador, 1: <NL> + EOI, 2: EOI), mientras que en VISA WRITE no podemos y se enviará uno compatible con IEEE-488.2

Si el instrumento que tenemos es compatible IEEE-488.2 no habrá problema, pero si no lo es debemos averiguar qué tipo de terminador requiere y suministrárselo, de lo contrario no entenderá que hemos acabado la comunicación.

Peticiones de servicio

El esquema de funcionamiento secuencial de la relación entre el controlador de un sistema de instrumentación y los instrumentos, en el que el controlador envía una orden de medida y espera la respuesta, es muy simple pero no siempre es el más conveniente.

Algunos instrumentos pueden tener tiempos de medida largos, o puede ser posible programar una secuencia de medidas temporizada de forma que el controlador no deba preocuparse del instrumento hasta que acabe o hasta que la medida esté disponible. En estos casos es útil disponer de un mecanismo que permita a los instrumentos notificar al controlador que se ha llegado a un determinado estado.

En un entorno IEEE-488, las interrupciones que pueden generar los instrumentos se llaman peticiones de servicio (*Service Request*) y hay una línea física en el bus dedicada a este menester (SRQ). El funcionamiento es muy similar al esquema de peticiones de interrupción en un entorno de microprocesador. En primer lugar hay que configurar al instrumento para que pueda activar SRQ cuando un determinado evento ocurra. Esto se hace programando los registros de estado del instrumento. En segundo lugar hay que definir en el controlador un mecanismo de respuesta en caso de petición de servicio. Genéricamente se puede hablar de dos categorías de mecanismos de respuesta: o bien guardamos la información de que ha llegado una petición de servicio en una cola que se va consultando periódicamente o bien activamos la ejecución de una rutina al recibir la interrupción. Dentro de VISA, a estos mecanismos se les conoce con *Queuing* y *CallBack*. Es necesario hacer notar que en LabVIEW, debido a la naturaleza del propio programa, cuya ejecución está controlada por el flujo de datos y puede ser paralela, el mecanismo de activación automática de una rutina no es posible, aunque se simula.

El primer paso a realizar cuando se sirve una petición de servicio es identificar al instrumento causante de la misma. La norma IEEE-488.1 establece dos mecanismos llamados *Serial Poll* y *Parallel Poll*. En el primero el controlador va preguntando secuencialmente a todos los instrumentos si han generado una interrupción hasta encontrar uno que responda afirmativamente. En un *Parallel Poll* se hace la pregunta a todos y contestan todos a la vez, usando cada uno una línea de datos. Obviamente no puede haber más de 8 instrumentos en este último caso.

Una vez determinado el causante de la petición se puede determinar el motivo de la misma leyendo sus registros de estado.

La norma IEEE-488.2 establece un modelo de registros de estado que se puede ver en la figura 3.4. El registro de estado (*Status Byte Register*) nos indica si el instrumento ha generado la petición (bit 6 a "1") y el motivo. Sólo hay dos bits asignados por la norma, el bit 5 asociado al Registro de Estado de Sucesos Normales (*Standard Event Status Register*) y el bit 4 asociado a la cola de mensajes de salida. El fabricante puede definir la funcionalidad de los otros bits. Para que la puesta a uno de estos bits genere una petición de servicio hay que habilitar la máscara del registro de estado (*Service Request Enable Register*).

El registro de estado de sucesos normales tiene 16 bits, 8 de los cuales están definidos en la norma. En la figura 3.4 pueden verse las condiciones que nos indica cada uno de ellos. Para que alguno de estos sucesos pueda activar el bit 5 del *Status Byte Register* también hay que habilitar una máscara.

Hay una serie de comandos básicos obligatorios en la norma IEEE-488.2 que permiten leer y escribir en estos registros:

- *STB? Devuelve el valor del Registro de Estado
- *SRE? Devuelve el valor de la máscara del Registro de Estado

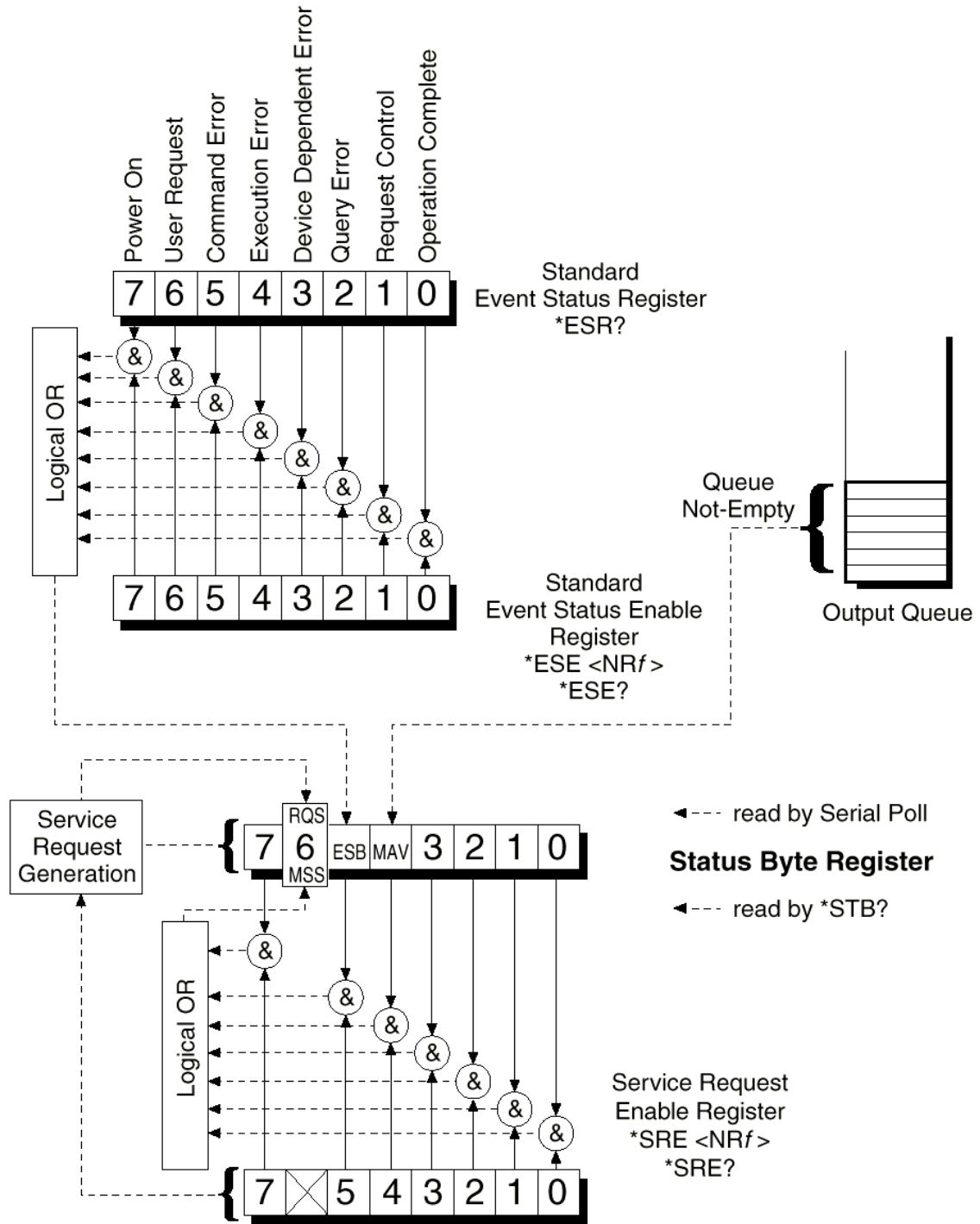


Fig. 3.4 Modelo de registros de estado especificado en la norma IEEE-488.2 junto con las órdenes básicas que permiten conocer y fijar su valor.

- *SRE ##** Fija el valor de la máscara del registro de estado. P.ej. ***SRE 32** habilita el bit 5 para que genere una petición de servicio
- *ESR?** Devuelve el valor del Registro de Estado de Sucesos Normales (RESN)
- *ESE?** Devuelve el valor de la máscara del RESN
- *ESE ##** Fija el valor de la máscara del RESN. P. ej. ***ESE 1** habilita el bit de *Operation Complete* para que active el bit 5 del Registro de Estado

Los instrumentos no compatibles con IEEE-488.2 también disponen de un Registro de Estado, similar al descrito más arriba, pero por lo general no aceptarán las ordenes descritas ni dispondrán de un Registro de Estado de Sucesos Normales. Habrá que consultar el manual de estos instrumentos para ver cómo se accede al registro de estado y cómo se habilitan las peticiones de servicio.

Peticiones de servicio en GPIB 488.2 y LabVIEW

Ya se ha mencionado que LabVIEW 4.1 no tiene la capacidad real de ejecutar un código específico cuando se recibe la señal SRQ (lo que en términos informáticos se conoce como una rutina de servicio de interrupción).

Para poder ejecutar el código deseado al recibir SRQ deberemos, en primer lugar, comprobar de forma periódica el valor de esta línea. Hay dos funciones en la librería GPIB 488.2 que nos permiten realizar esta tarea: WAITSRQ y TESTSRQ. La primera detiene la ejecución del programa hasta que se activa SRQ o bien se produce un *TimeOut*. La segunda simplemente comprueba el valor de la línea y devuelve un valor booleano verdadero o falso.

Hay muchas formas de utilización de estas dos funciones para realizar lo que nosotros deseamos. Pueden verse algunos ejemplos en una nota de aplicación de National Instruments Inc. en <http://www.natinst.com/pdf/instrupd/appnotes/an126.pdf>.

Como indicación general podemos decir: WAITSRQ puede ser útil si conocemos de antemano el tiempo que tardará el equipo en producir la interrupción. No es aconsejable usar *TimeOuts* infinitos porque el procesador puede quedar bloqueado. TESTSRQ puede usarse en un bucle, infinito, hasta que se active SRQ, con lo cual funcionaría de forma similar a WAITSRQ. No obstante, dada la capacidad de ejecución en paralelo de LabVIEW, mientras se ejecuta este código se pueden hacer otras cosas. Aplicado de esta forma simularía una rutina de servicio de interrupción, aunque la eficiencia es mucho más baja.

Una vez hemos detectado la activación de SRQ debemos identificar al causante de ella. También nos ofrece la librería GPIB 488.2 dos funciones para ello: ALLSPOLL y FINDRQS. La primera realiza un *Serial Poll* a todos los instrumentos de la lista especificada y devuelve un vector con los Registros de Estado de todos ellos. Con esta información el usuario puede identificar al (o los) culpable(s) de la petición de servicio. FINDRQS realizará un *Serial Poll* de los dispositivos en la lista que se le proporcione, devolverá la dirección del primero que encuentre con el bit 6 del Registro de Estado activo y enviará un **SDC** (Selected Device Clear) a todos los dispositivos de la lista, borrando la información de todos los Registros de Estado.

Esta función sólo debería usarse si se está seguro que un único instrumento ha podido realizar una petición de servicio.

Una vez leído el registro de estado debe leerse también el Registro de Estado de Sucesos Normales (***ESR?**) para conocer (y borrar) la condición que ha provocado la petición de servicio. Si no se hiciera esto, la condición permanecería y el instrumento volvería a generar una petición de inmediato.

Peticiones de servicio en VISA y LabVIEW

Para gestionar las peticiones de interrupción y otras situaciones anormales en el sistema, en VISA se define el concepto de suceso (*event*). Un suceso puede ser una petición de servicio GPIB, o una activación de una interrupción VXI o incluso una excepción del procesador.

Existen dos mecanismos de tratamiento de sucesos, que en terminología VISA se conocen como *Queuing* y *Callback*. Los funciones que permiten descubrir si se ha producido un evento determinado asociado a estos dos mecanismos se conocen como espera síncrona y espera asíncrona, respectivamente. Es un funcionamiento paralelo al comentado para GPIB 488.2. Aunque en LabVIEW hay una función llamada WAIT ON EVENT ASYNCHRONOUSLY, de hecho es una simulación realizada a base de ejecutar en paralelo con el resto del programa la función WAIT ON EVENT dentro de un bucle (similar a la utilización de TESTSRQ que se ha comentado en el apartado anterior).

Hay dos diferencias fundamentales, a nivel conceptual, entre el tratamiento de peticiones de servicio usando funciones GPIB 488.2 y VISA. En primer lugar, para que el gestor de recursos de VISA detecte algún suceso debemos habilitarlo explícitamente con la función VISA ENABLE EVENT. En segundo lugar, cuando se detecta un evento se abre una sesión para el tratamiento del mismo, que hay que cerrar una vez se han realizado las operaciones pertinentes. Esta sesión es similar a la sesión que se abre para comunicarse con un recurso del sistema.

Dado que las peticiones de servicio del bus GPIB son un caso particular bastante usado en la práctica, National Instruments ha creado un VI para LabVIEW específico para tratar este tipo de suceso. Este VI se basa en la ejecución en un bucle de la función VISA WAIT ON EVENT hasta que se detecta SRQ. Esta función también lee el registro de estado del instrumento y cierra la sesión de tratamiento del suceso de forma automática. En la figura 3.5 se ve el diagrama de un VI que recoge lecturas de un instrumento usando la función VISA WAIT FOR SRQ.

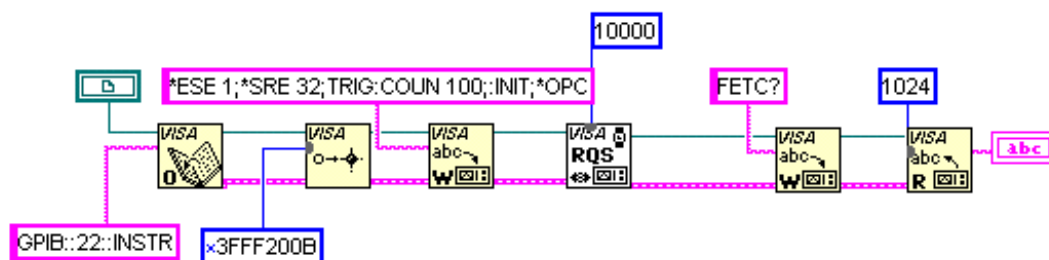


Fig. 3.5 Utilización de la detección de eventos de VISA para esperar la finalización de una serie de medidas y después leer su valor.

En primer lugar se abre una sesión VISA a un instrumento GPIB en la dirección 22. A continuación se habilita la detección de eventos del tipo SRQ, mediante el código 0x3FFF200B. Después programamos al instrumento, habilitando la generación de peticiones de servicio y *operation complete* y damos las ordenes de medida pertinentes.

La función WAIT FOR SRQ está construida a base de un bucle que va mirando el valor de la línea SRQ del bus. Debemos especificar durante cuánto tiempo el bucle esperará la petición, mediante la variable *timeout*, que en este caso se ha puesto a 10 s.

Mientras esta función se ejecuta, otros VI en paralelo con ella también se podrían estar ejecutando.

Cuando la función finaliza pasamos a leer los datos obtenidos por el instrumento, con VISA READ.

Al igual que se ha comentado en el apartado anterior, debemos borrar la condición de petición de servicio en el registro de sucesos normales, ya sea leyéndolo o borrando explícitamente su contenido, de lo contrario la condición de petición permanecerá activa.

Trabajo previo

Compatibilidad con IEEE-488.2

Determinar, para todos los instrumentos del puesto de trabajo, la configuración de los registros de estado (¿son compatibles IEEE-488.2?, ¿dispone de registros adicionales?, etc.) y la forma de acceder a los mismos.

Generación de peticiones de servicio

La cadena de programación que aparece en la figura 3.5 está pensada para el multímetro digital HP34401A. Explicar brevemente el significado de cada una de las unidades de programación, especialmente *OPC

VISA v.s. GPIB-488.2

¿Cual es la diferencia fundamental, aparte del uso de las funciones VISA y GPIB 488.2, entre los diagramas de la figura 3.5 y el de la página siguiente?. Explicar brevemente el funcionamiento del VI de la página siguiente.

Trabajo de laboratorio

Realización automática de medidas con el multímetro

El multímetro HP34401A dispone de una memoria interna donde puede almacenar medidas y también dispone de una unidad del cálculo que le permite hacer operaciones con estas medidas.

Usando las funciones de la librería GPIB-488.2 o las de VISA, programar el multímetro para que realice una serie de 100 medidas, de tensión alterna, con la máxima resolución, las guarde en la memoria interna y calcule la media. Una vez finalizada la medida, leer el valor de la media, el valor máximo y el mínimo.

Detección de la finalización de las medidas

Usando las funciones VISA, programar el multímetro para que realice la misma medida que en el apartado anterior. Disponer el multímetro para que genere una petición de servicio al finalizar y detectar la petición de servicio usando un esquema parecido al de la figura 3.5. Una vez detectada la petición de servicio, obtener los mismos datos que en el apartado anterior.

Para comprobar que esta función simula efectivamente el mecanismo *Callback*, realizar en paralelo un programa que p.ej. actualice un indicador en la pantalla.

Trabajo opcional

Gestión de peticiones de servicio con múltiples instrumentos

Usando el VI de la página siguiente, programar al menos dos instrumentos para que generen una petición de servicio. Cuando se reciba la petición, indicar en la pantalla quién ha sido el causante y volver a disponer el instrumento en las condiciones iniciales.

Práctica 4

Utilización de controladores de instrumentos. Errores sistemáticos

Objetivos

El objetivo de la práctica es doble. Por un lado, familiarizarse con el uso de controladores de instrumentos, y por otro, evaluar los errores sistemáticos de los instrumentos utilizados.

Para ello se programarán, mediante los controladores de instrumentos disponibles en las librerías de LabVIEW, los instrumentos necesarios (generador de funciones y multímetro) y se realizarán medidas de amplitud y de frecuencia de las señales senoidales generadas. Las diferencias entre los valores programados y medidos se compararán con la incertidumbre esperada deducida de las especificaciones de los equipos. Con este método podría detectarse si hay algún equipo con errores sistemáticos superiores al especificado.

Primero se presentará una breve introducción a la estructura de los controladores de instrumentos en LabVIEW y posteriormente se planteará el sistema de generación y medida para evaluar la incertidumbre total esperada bajo diversas condiciones.

Controladores de instrumentos

Un controlador, o en Inglés *driver*, de instrumento es un conjunto de funciones (VI) que permiten el control de un instrumento específico sin tener que recurrir a las funciones básicas de la interfase de control o de las funciones de VISA. LabVIEW dispone de una librería en el panel de funciones para los controladores de instrumentos. Esta librería se tiene que crear para cada entorno específico con los controladores de los instrumentos que se desean controlar. Los controladores pueden ser suministrados por los fabricantes de los equipos o por el fabricante del entorno de programación, en este caso National Instruments a través de su servidor de web (<http://www.natinst.com>) o de CDROM.

Modelo de interfase externa de los controladores

Un controlador de instrumentos en LabVIEW interactúa con el resto del sistema según el modelo general de la figura 4.1. El cuerpo funcional es el conjunto de funciones que controlan el instrumento. Está formada por un conjunto de VI que están desarrollados con diagramas de bloques por lo que pueden ser analizados o modificados con las herramientas de LabVIEW.

Los front panels de cada VI (Interfase de desarrollo) pueden ser utilizados para hacer pruebas de forma interactiva y facilitar así la depuración de los errores de programación.

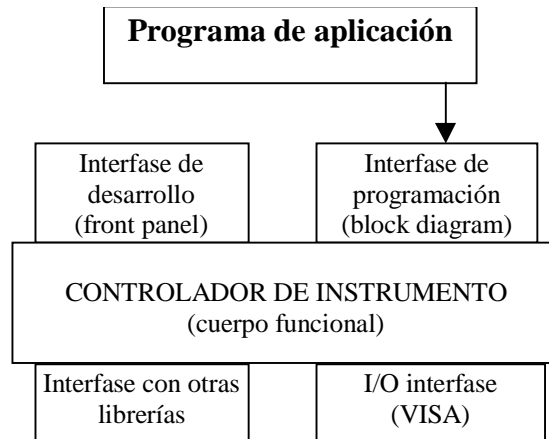


Fig. 4.1 Modelo de comunicación de un controlador con el entorno.

Los iconos y los conectores de cada uno de los VI que forman el controlador constituyen las entradas y salidas que comunican el controlador con el programa general de la aplicación. De esta forma el desarrollo de un programa de aplicación consistirá en la unión de diversos subVI extraídos (y modificados si es necesario) de las librerías de controladores de instrumentos y las funciones auxiliares de procesamiento y presentación de datos.

Los subVI que componen la librería de un controlador están a su vez formados por subVI de control de la interfase física de comunicaciones a través de funciones VISA y de funciones de otras librerías como por ejemplo de análisis y procesamiento de datos.

Estructura interna de los controladores

La estructura de un controlador en LabVIEW responde al esquema de la figura 4.2. Todos los controladores de instrumentos tendrían que responder a la misma estructura para facilitar el desarrollo de programas. LabVIEW contempla dos tipos de funciones para un controlador: las funciones de aplicación (Application functions) y las funciones básicas (Component functions). Las funciones de aplicación representan el nivel jerárquico más alto de programación, permitiendo con un solo VI el control de las funciones y medidas más habituales de un

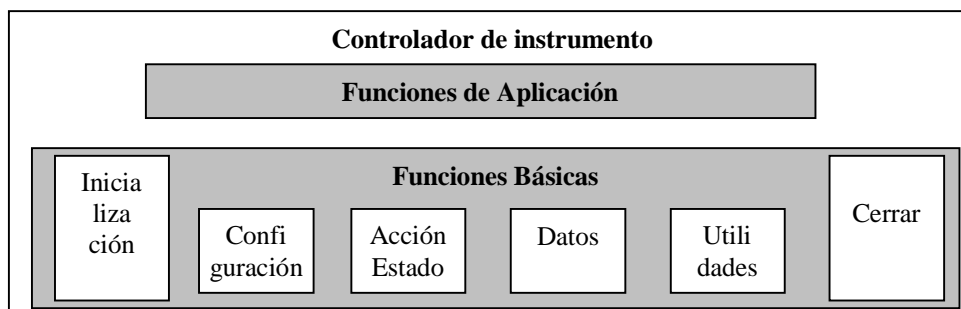


Fig. 4.2 Componentes funcionales de un controlador de instrumento

instrumento. Las funciones de aplicación están orientadas a poder realizar las medidas más habituales de forma casi inmediata, pero también pueden servir de ejemplo para realizar aplicaciones parecidas que se adapten a las necesidades particulares del usuario.

Las funciones básicas de control son un conjunto de subVI que permiten programar las distintas opciones de cada instrumento. Las funciones básicas se dividen en seis categorías: Inicialización, configuración, acción/estado, datos, utilidades y cerrar (close).

Las funciones de inicialización y de cerrar son obligatorias para todos los controladores. La función de inicialización establece la comunicación con el instrumento y debe ser la primera función a ejecutar. Esta función puede además configurar el instrumento en un estado inicial concreto (default). La función de cerrar corta la comunicación con el instrumento liberando los recursos del sistema utilizados para ella.

Las funciones de configuración permiten programar el equipo de la forma más adecuada para realizar posteriormente una medida o generar una señal.

Son las funciones de acción las que causan que el instrumento inicie o termine una acción, teniendo en cuenta la configuración que se ha enviado previamente. Estas funciones pueden incluir armar el sistema de disparo o iniciar una acción (trigger). Las funciones de estado permiten conocer cual es la situación del instrumento y si hay operaciones pendientes de ejecución.

Las funciones de datos son las que permiten transferir datos desde o al instrumento. Pueden ser desde la simple transmisión de un dato numérico a la transmisión de todo un vector de una o más dimensiones.

Los subVI de utilidades realizan funciones auxiliares no soportadas en los anteriores grupos. Por ejemplo: *resets*, tratamiento de errores, test, calibraciones, etc.

En los controladores de instrumentos más profesionales se incluye un VI (Tree VI) que permite visualizar todas las funciones que componen el controlador y su jerarquía. También es recomendable que se incluya el llamado "Starter VI" que sirve para poder comprobar la funcionalidad de nuestro sistema sin tener que realizar ningún programa y puede servir de ejemplo para futuras aplicaciones.

Trabajo previo

Sistema a diseñar. Evaluación de errores

El sistema estará compuesto por el generador de funciones al que se conectará y el multímetro digital. La forma de onda, la frecuencia y la amplitud de la señal generada serán programables desde el panel del VI a diseñar. La frecuencia y la amplitud (valor eficaz) de esta señal se medirán con el multímetro.

1. A partir de las especificaciones del generador de funciones evaluar la incertidumbre en la frecuencia y en la amplitud de la señal generada a las siguientes frecuencias (con una amplitud de 1 V_{ef}): 10 Hz, 10 kHz y 1 MHz. (Nota: suponer que los equipos se calibran una vez al año)
2. Determinar el intervalo de incertidumbre en frecuencia y amplitud al medir con el multímetro la señal anterior a las frecuencias dadas con una resolución de 6 1/2 dígitos y con el filtro rápido de acoplo en AC.

Trabajo de laboratorio

1. Diseñar un VI que controle el generador de funciones y el multímetro utilizando los controladores de instrumentos de las librerías de LabVIEW. El panel tiene que tener los controles para modificar la frecuencia, la amplitud y la forma de onda (senoidal, triangular o cuadrada) y mostrar los siguientes valores: frecuencia y amplitud programadas, frecuencia y amplitud medidas con el multímetro y diferencias entre ellas. Se recomienda partir de algún ejemplo simple de utilización del generador de funciones y editarlo para incluir las funciones de medida del multímetro y las comparaciones.
2. Programar una señal senoidal de 1 Voltio eficaz y una frecuencia de 10 kHz. ¿Están los resultados dentro del margen esperado según las especificaciones de los equipos?
3. Además del posible error sistemático ¿hay errores aleatorios? Realizar varias medidas para ver si son del mismo orden que los sistemáticos. No realizar medidas estadísticas, estas se harán en la siguiente sesión.
4. Con los mismos valores de amplitud y de frecuencia cambiar la forma de onda a cuadrada. ¿Cuál es el valor medido por el multímetro?, ¿coincide con el valor eficaz de la señal? ¿Qué tipo de conversor alterna continua tiene el multímetro?
5. Realizar las medidas con una senoide de 10 Hz y de 1 MHz ¿Están los resultados dentro del margen esperado según las especificaciones de los equipos?, ¿cómo se explican los resultados en ambos casos?

Trabajo opcional

Utilizando el VI anterior como base crear un VI que haga un barrido de frecuencias entre una frecuencia mínima y una máxima con un paso seleccionable (octavas o décadas).

Práctica 5

Procesado automático de medidas. Análisis y reducción de ruido

Objetivo

El objetivo de esta práctica es analizar y reducir los errores aleatorios (ruido) que aparecen en un sistema de medida automático. En concreto se estimarán la media y la desviación tipo de las medidas y se presentará el histograma de las medidas adquiridas. Si es posible se especificará la incertidumbre de la medida calculando un intervalo de confianza.

A partir del análisis del histograma obtenido se identificará la fuente de ruido y se realizarán acciones para reducir su efecto sobre las medidas.

Sistema de medida

Como ejemplo de una aplicación muy sencilla, pero en la que ya aparecen problemas de ruido e interferencias, se ha tomado la medida del valor de una resistencia mediante un multímetro digital.

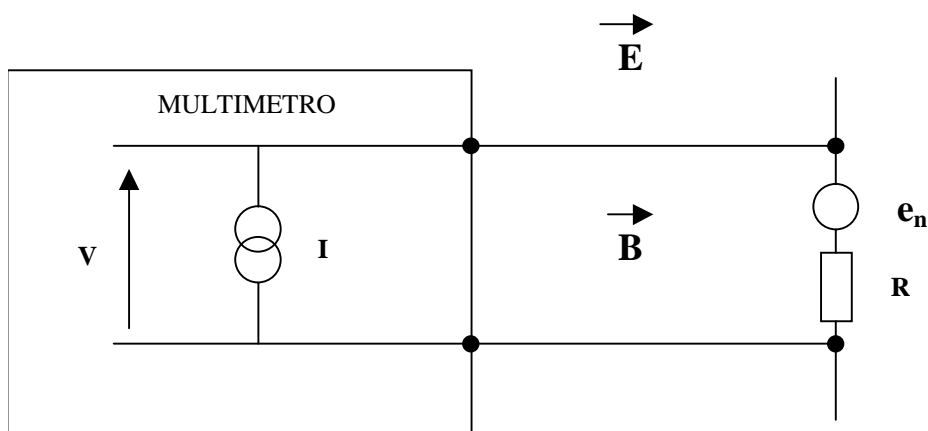


Fig. 5.1 Sistema de medida con algunas fuentes posibles de ruido e interferencias

El objetivo de la medida será estimar, con la menor incertidumbre posible, el valor de una resistencia de 1 M Ω de valor nominal.

El sistema de medida estará constituido, en principio, por el multímetro, los cables de conexión a la resistencia y el sistema de control. Como la resistencia es de un valor elevado se utilizará el método de medida a dos hilos y no a cuatro. El multímetro estima la resistencia como la relación entre la tensión continua medida (V) y la corriente continua inyectada (I) que es de valor conocido. En la figura 5.1 se presenta el sistema de medida y tres posibles causas que pueden incrementar la componente aleatoria de las medidas.

Análisis de las posibles fuentes de ruido e interferencias

La primera fuente de una señal aleatoria puede ser el ruido propio de la resistencia (e_n). El ruido térmico se supone que es blanco y gaussiano. La densidad espectral de tensión equivalente de ruido de una resistencia es

$$\sqrt{4kTR} \text{ (V}/\sqrt{\text{Hz}})$$

siendo aproximadamente de **130 nV**/ $\sqrt{\text{Hz}}$ para 1 M Ω a una temperatura de 300 K

Otras fuentes de ruido que pueden afectar a la medida de V son los campos magnéticos (B) y eléctricos (E) que existan en nuestro entorno de trabajo. Las principales fuentes de estos campos electromagnéticos serán la red de distribución de energía eléctrica (50 Hz), los aparatos a ella conectados y los sistemas de telecomunicaciones de nuestro entorno.

Todas estas fuentes tienen un efecto aditivo sobre la señal de tensión V pero el efecto sobre la lectura (valor estimado de R) dependerá de su espectro de potencia y del ancho de banda equivalente con el que se mida la tensión V. Para el caso particular de medida de resistencias con un multímetro digital, la función de transferencia y por tanto el ancho de banda equivalente dependerá del tiempo de integración que utilice el multímetro para hacer la medida.

Trabajo previo

1. A partir de las especificaciones del multímetro determinar la incertidumbre provocada por los errores sistemáticos al medir una resistencia de un valor próximo a 1 M Ω .
2. Calcular cual es el efecto de la tensión equivalente de ruido de la resistencia si el tiempo de integración es de 4 ms y la corriente inyectada es la que especifica el instrumento en la escala de 1 M Ω (consultar el manual). Expresar el resultado como la desviación tipo del resultado en Ohmios.
3. ¿Cuál es la función de densidad de probabilidad de la lectura si suponemos que sólo hay este ruido?
4. Si suponemos que el campo electromagnético más importante es el de 50 Hz y este inyecta en nuestro circuito una señal aditiva senoidal de 10 mV de amplitud. ¿Qué desviación tipo, expresada en ohmios tendrá la lectura?
5. ¿Qué forma aproximada tiene la función de densidad de probabilidad de la lectura para este caso? (despreciar el ruido propio de la resistencia)

6. Si el tiempo de integración del multímetro se programa para que sea de 200 ms ¿cuál es la desviación tipo de la salida debido a la señal senoidal?

Trabajo de laboratorio

1. Diseñar un VI, partiendo de la librería del multímetro, que mida resistencia a dos hilos, se pueda programar el tiempo de integración (directamente o mediante la selección del número de dígitos, ya que están relacionados¹) y seleccionar el número de medidas consecutivas a realizar.

El panel del VI debe presentar:

- el vector de medidas con un visualizador numérico para vectores o de forma gráfica
- el valor medio de las medidas realizadas
- la desviación tipo de las medidas

Para realizar las funciones estadísticas utilizar la librería de Probabilidad y estadística que está dentro de la librería de Análisis.

2. Conectar la resistencia de 1 M Ω de valor nominal al multímetro utilizando dos cables unifilares y bananas. Realizar un conjunto de 100 medidas con un tiempo de integración de 4 ms.
3. ¿Está el valor medio estimado dentro del margen de incertidumbre calculado para los errores sistemáticos? Si no es así proponer alguna hipótesis.
4. ¿Coincide la variancia calculada con la que puede provocar únicamente la tensión propia de ruido de la resistencia?
5. Para averiguar si los campos de 50 Hz son los responsables de la variancia obtenida ampliar el VI diseñado con una gráfica que represente el histograma de las medidas. Realizar un conjunto amplio de medidas para que el histograma represente aproximadamente la función de distribución de probabilidad de las medidas. ¿El histograma obtenido se aproxima más a una gaussiana o al esperado de una señal senoidal?
6. ¿Se pueden calcular los intervalos de confianza y de tolerancia con los resultados disponibles? ¿Por qué?
7. Intentar reducir el efecto de las interferencias minimizando la longitud de los cables, reduciendo el bucle que forman y/o apantallando. El reducir la fuente de interferencias es otro buen método, pero no es fácil (o muy útil) el apagar todos los equipos del laboratorio.
8. ¿Se pueden calcular los intervalos de confianza y de tolerancia con los resultados disponibles? ¿Por qué?
9. Comprobar los resultados si se aumenta el tiempo de integración a 200 ms. ¿Es ahora posible calcular el intervalo de confianza? ¿Por qué? Tener en cuenta que al medir durante un tiempo prolongado puede haber derivas debido a cambios en la temperatura del resistor (el coeficiente térmico de una resistencia de carbón es de unos $1500 \cdot 10^{-6}/^{\circ}\text{C}$)

¹ El driver que suministra National Instruments junto con LabVIEW sólo acepta la programación del número de dígitos. Al seleccionar una resolución, el multímetro se configura para hacer medidas con el tiempo de integración más corto posible para esa resolución -- consultar las combinaciones posibles en el manual del instrumento.

Trabajo opcional

Modificar el VI para presentar la medida y el intervalo de incertidumbre total debido a los errores tanto sistemáticos como aleatorios

Práctica 6

Diseño de un analizador de espectros virtual

Objetivo

El objetivo de esta práctica es crear, mediante el uso del osciloscopio digital y las librerías de análisis de LabVIEW, un analizador de espectros virtual.

Para programar el osciloscopio se utilizará el controlador suministrado en la librería de controladores de instrumentos

Las librerías de análisis de LabVIEW

Dentro de la paleta de funciones encontramos las librerías de análisis de LabVIEW. En este caso aparecen 10 librerías de análisis ya que están instaladas las librerías de análisis avanzado. Las librerías básicas sólo incluyen análisis estadístico, álgebra lineal y cálculo numérico. Las librerías avanzadas incluyen VI para los siguientes temas:

- Generación de señal (senoidal, ruido, formas de onda arbitrarias, etc.)
- Procesado digital (análisis temporal y frecuencial, transformadas: DFT, Hilbert, etc.)
- Medidas (estimaciones de espectros y otras medidas en el espacio frecuencial)
- Filtros (IIR, FIR, no lineales, etc.)
- Ventanas (Hanning, Hamming, Blackman, coseno alzado, etc.)
- Probabilidad y estadística (media, desviación tipo, funciones de probabilidad, etc.)
- Aproximación a curvas e interpolación
- Álgebra lineal (para vectores y matrices reales o complejos)
- Operaciones con vectores y matrices
- Otros métodos numéricos (raíces de polinomios complejos, integración numérica, detección de pico, etc.)

Las librerías de Medidas están orientadas al desarrollo de aplicaciones para medidas en el dominio frecuencial basadas en transformadas de Fourier. Estas librerías son útiles para desarrollar de forma rápida instrumentos virtuales de análisis en el dominio frecuencial (tales como analizadores de espectros y analizadores de redes) basándose en un sistema de adquisición de datos. En los ejemplos existentes en las librerías (\examples\analysis\measures\measxmpl.llb) puede verse el uso típico de los VI y su interconexión.

Para ver las posibilidades de estas librerías y realizar una aplicación propia en el menor tiempo se recomienda abrir el ejemplo de analizador de espectros (SIMPLE SPECTRUM ANALIZER (sim) que se encuentra en la librería: MEASXMPL.LLB. En este ejemplo (ver figura) la señal se obtiene

con un generador de funciones simulado, la señal es enventanada, pudiéndose escoger el tipo de ventana, y posteriormente se calcula el espectro de potencia utilizando una única realización. El VI de escalado permite seleccionar las unidades de presentación y se incluye, además, un detector del pico mayor del espectro que nos presenta la frecuencia a la que ocurre y su potencia estimada.

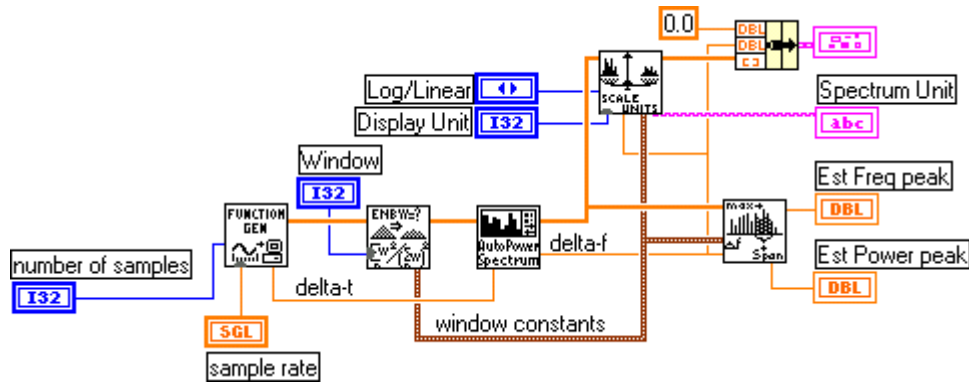


Fig. 6.1 Ejemplo de analizador de espectros simulado

Estimación del Espectro de potencia

El espectro de potencia de una señal se define como:

$$S_{xx}(f) = \lim_{T \rightarrow \infty} \frac{1}{T} E \left[|FT[x(t, T)]|^2 \right]$$

Donde T es la longitud del registro, E es la esperanza matemática y FT la transformada de Fourier. Si la señal es de longitud finita, se discretiza la señal y se aproxima la esperanza de la transformada por una sola realización, se obtiene una estimación del espectro (expresado en V²) de potencia dado por:

$$\hat{S}_{xx}(k) = \frac{1}{N^2} |DFT[x(n)]|^2$$

Este estimador es inconsistente (su variancia no decrece al aumentar N):

$$\text{var} [\hat{S}_{xx}(k)] \approx S_{xx}^2(f)$$

por lo que normalmente se procede a utilizar un promediado de espectros de la forma:

$$\hat{S}_{xxn_d}(k) = \frac{1}{n_d N^2} \sum_{i=1}^{n_d} |DFT[x_i(n)]|^2$$

y en este caso la variancia del estimador si que se reduce al aumentar las realizaciones:

$$\text{var} [\hat{S}_{xxn_d}(k)] = \frac{1}{n_d} \text{var} [S_{xx}(f)]$$

Trabajo previo

Osciloscopio

1. Buscar las siguientes especificaciones del osciloscopio:
 - Ancho de banda
 - Frecuencia de muestreo máxima
 - Resolución del conversor A/D
 - Longitud máxima de memoria de traza
2. ¿Permite el osciloscopio poner algún filtro para limitar el ancho de banda de los canales verticales y evitar así posible aliasing?
3. ¿Cuál será la máxima frecuencia que se podrá presentar en el espectro si sólo se utiliza un canal del osciloscopio en modo de disparo único?
4. ¿Y si la adquisición de la señal se realiza mediante muestreo repetitivo?

Trabajo de laboratorio

1. Modificar el ejemplo de analizador de espectros para que pueda seleccionarse, por el panel del instrumento, el número de realizaciones a promediar para disminuir el ruido del estimador. Salvar el nuevo VI con otro nombre en el disco de usuario. El promediado de espectros puede realizarse mediante el uso de un registro de desplazamiento (*shift register*) dentro del bucle que acumule los resultados parciales y dividiendo por el número de medidas a la salida del bucle. Acordarse que es imprescindible inicializar a cero el *shift register* antes de iniciar el bucle.
2. Comprobar el efecto sobre el ruido del estimador al incrementar el número de promediados, para ello añadir ruido blanco a la señal simulada con una relación de 10 entre la amplitud de la señal y la del ruido.
3. Realizar, utilizando las librerías de controladores de instrumentos, una aplicación que permita obtener un registro de 2000 muestras con la señal digitalizada por el osciloscopio en uno de sus canales. Se recomienda que el panel tenga los siguientes controles e indicadores:
 - Controles: frecuencia de muestreo, V/DIV,
 - Indicadores: presentación gráfica de la señal adquiridaPuede usarse un esquema parecido al de la figura 2.
4. Definir un subVI para el control del osciloscopio e integrarlo en el analizador de espectros diseñado previamente.
5. Conectar un generador de funciones al osciloscopio y verificar el funcionamiento para distintos tipos de señales (el control del generador puede hacerse manualmente). En concreto verificar las limitaciones debidas a aliasing.

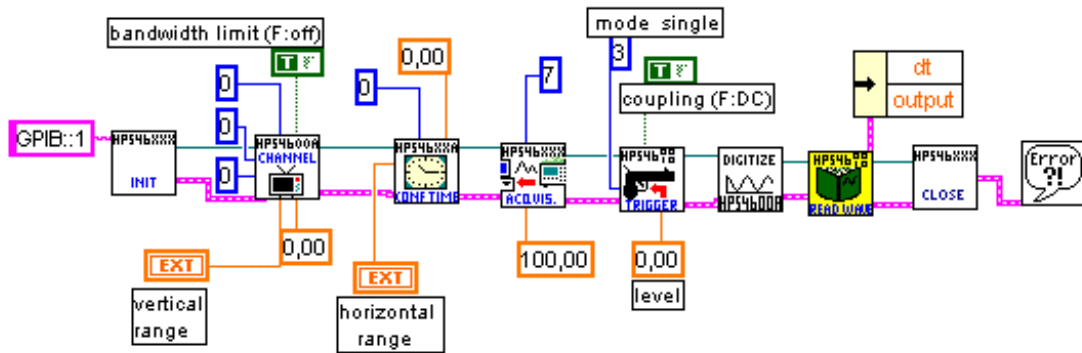


Fig. 6.2 V.I. que configura el osciloscopio, ordena la realización de una adquisición y obtiene los datos adquiridos.

Trabajo opcional

La mayoría de analizadores de espectros comerciales incorporan un suavizado de datos a la salida del estimador de potencia denominado filtro de vídeo. Realizar este suavizado mediante un filtro paso bajo a la salida de la estimación de potencia. El filtro debe poderse activar y desactivar a través del panel del analizador.

Integrar al conjunto anterior el control del generador de forma automática desde LabVIEW realizado en prácticas previas.

Práctica 7

Medida de la respuesta frecuencial de un filtro de audio

Objetivo

El objetivo de la práctica es crear, mediante un osciloscopio digital de doble canal y el LabVIEW un analizador de redes que permita medir la respuesta frecuencial de un sistema lineal en régimen permanente senoidal. Además se analizarán las fuentes de error que influyen en las medidas y su propagación al resultado final.

Para las pruebas se dispondrá de un filtro paso banda para aplicaciones de audio frecuencia. Además de la respuesta frecuencial, en módulo y fase el sistema deberá medir:

- Frecuencias de corte a -3dB
- Retardo de grupo máximo

Medidas con señales senoidales

La medida de la respuesta frecuencia de un sistema se puede realizar de diferentes maneras. La más simple es excitar el sistema con una señal senoidal y observar la salida. Si se realiza un barrido de frecuencia y se mide la amplitud de salida y el desfase, se obtiene la respuesta frecuencial del sistema. Éste es el principio en que se basan los analizadores de redes de radiofrecuencia. Sin embargo, existen equipos de baja frecuencia que como señal de excitación utilizan una fuente de ruido blanco. La determinación de la función de transferencia se realiza en este caso a partir de las densidades espectrales de potencia a la entrada y salida y del espectro cruzado. La ventaja de estos sistemas radica fundamentalmente en la velocidad de obtención de la función de transferencia. Además, en algunos sistemas no se puede inyectar una excitación senoidal de gran amplitud porque podría introducir distorsiones o llevarlo a la inestabilidad. El precio que se paga es una mayor complejidad del equipo y mayor incertidumbre en la estimación. Para solventar el problema del error incorporan la función de promediado espectral, que se han visto en la práctica anterior.

El método propuesto para esta práctica se basa en la excitación senoidal. El instrumento virtual que realizará la medida hará uso del generador de funciones, el osciloscopio digital y el multímetro digital. Las medidas de desfase se realizarán con el osciloscopio (ver figura 7.1). Para las medidas de amplitud se podrá utilizar el osciloscopio o el multímetro indistintamente.

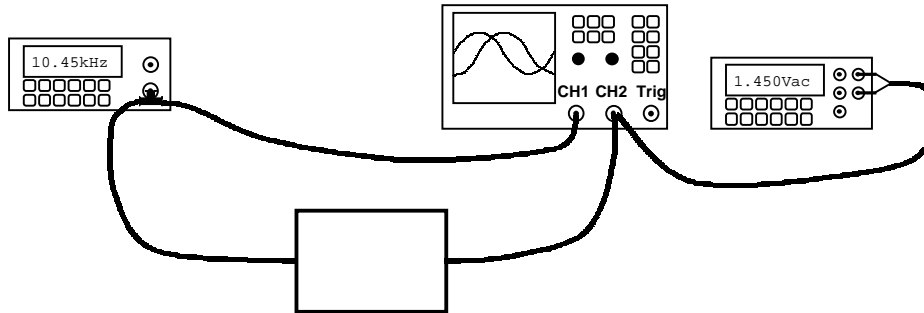


Fig. 7.1 Conexión de los instrumentos y el filtro para la medida de la función de transferencia

Trabajo previo

Una vez decidido el método de medida que se va a utilizar hay que analizar cuales son las posibles fuentes de error en las medidas y como afectarán al resultado final.

Diseño del filtro

1. Calcular el valor de los componentes de un filtro RC pasivo paso banda de segundo orden, con una frecuencia de corte paso alto de 20 Hz y una frecuencia de corte paso bajo de 20 kHz.
2. Estimar los errores en las frecuencias de corte teniendo en cuenta que la tolerancia de los condensadores es del 10% y la tolerancia de las resistencias del 5%.

Medida del módulo

Para medir la amplitud de señales con el osciloscopio se puede utilizar la función de medida automática que dispone el osciloscopio. Las medidas pueden ser de amplitud de pico a pico o pico, o bien de valor eficaz.

1. Determinar la expresión para el error en la medida del módulo de la función de transferencia debido al ruido de cuantificación del osciloscopio y errores del canal vertical cuando se mide en ambos canales. ¿Cuál es el mejor método de medida de tensión para determinar el módulo de la función de transferencia con el osciloscopio?.
2. Si se mide sólo la señal a la salida del filtro con el osciloscopio y se consideran los errores del generador de funciones, ¿cómo cambia el error en la medida?
3. ¿Cuál deberá ser la amplitud mínima del generador de funciones para que el error sea menor del 3%, en la banda de paso? (suponer un filtro de ganancia unidad en la banda de paso y un valor eficaz de ruido de 1 mV_{ef})
4. Si para medir la amplitud de salida del filtro se utiliza el multímetro digital, ¿cuál será el error total debido al ruido anteriormente especificado y a los errores del multímetro? Expresar dicho error a las frecuencias de 20 Hz, 2 kHz y 50 kHz.

Medida de la fase y el retardo de grupo

El osciloscopio digital HP5400A permite medir de forma automática el desfase entre los dos canales. Para ello mide el periodo de la señal aplicada al canal 1 y el retardo entre el paso por cero con pendiente positiva de los dos canales. Por tanto, la base de tiempos deberá estar configurada para que haya como mínimo un ciclo de señal en pantalla con dos pasos por cero

con pendiente positiva en el canal 1. Si no se cumple esta condición el osciloscopio devolverá como resultado de la medida el valor 9.9E37, indicando que no ha sido posible realizar la medida.

Al medir el paso por cero de una señal muestreada y cuantificada contribuyen diversas causas al error en la estimación. Las tres que tienen mayor importancia son: la incertidumbre en el paso por cero debido al muestreo, al ruido presente en la señal y al efecto de la cuantificación.

1. ¿Cuál será el error absoluto en grados en la medida del desfase debido al muestreo de la señal? Considerar que la memoria de pantalla son 500 muestras y aparece un ciclo y medio de señal.
2. ¿Cuál será el máximo error en la medida del desfase debido al ruido con una probabilidad del 95%? Considerar la relación señal a ruido a la entrada del filtro infinita y de 40 dB a la salida.
3. ¿Cuál será el error absoluto en grados en la medida del desfase debido a la cuantificación de la señal? Considerar que la memoria de pantalla son 500 muestras, se ha configurado la base de tiempos para tener un ciclo y medio de señal y que se ha ajustado la sensibilidad vertical para que la señal ocupe todo el margen dinámico del convertidor A/D (8bits).
4. Para reducir el error en la medida se decide reducir la base de tiempos para poder medir con mayor resolución la diferencia temporal entre pasos por cero de las dos señales, τ , y luego calcular el desfase como $\phi=2\pi f\tau$. ¿Cuál será ahora el error el error en la medida de un desfase de 90° ?

El retardo de grupo para un sistema lineal se define como:

$$\tau(f) = -\frac{1}{2\pi} \frac{d\phi(f)}{df}$$

Al realizar las medidas de la fase de la función de transferencia a frecuencias discretas, el retardo de grupo se puede estimar como:

$$\tau(f) = -\frac{1}{2\pi} \frac{\phi(f_2) - \phi(f_1)}{f_2 - f_1}$$

Siendo $\phi(f_1)$ y $\phi(f_2)$ los desfases medidos, en radianes, para las frecuencias f_1 y f_2 .

Trabajo del laboratorio

Para la creación del instrumento virtual partiremos de un ejemplo ya creado, `freqresp.llb`, que se encuentra en el directorio `soft\LabView-4\examples\apps`. En la figura 7.2 puede verse el diagrama del VI *Frequency response*.

1. Montar el filtro en placa de pruebas, tener en cuenta los efectos de carga de una etapa sobre la siguiente.
2. Crear un subVI que programe el generador de funciones con una forma de onda senoidal de frecuencia y amplitud variables.

3. Crear otro subVI que haga una medida de amplitud en los dos canales con el osciloscopio y mida el desfase entre las dos señales. El subVI deberá ajustar de forma automática la sensibilidad de los dos canales y la base de tiempos.
4. Incorporar los dos subVI's al esquema de la figura 7.2.
5. Añadir un subVI que a partir de la respuesta frecuencial en módulo y fase determine las frecuencias de corte a -3dB, el rizado en la banda de paso y el retardo de grupo máximo.

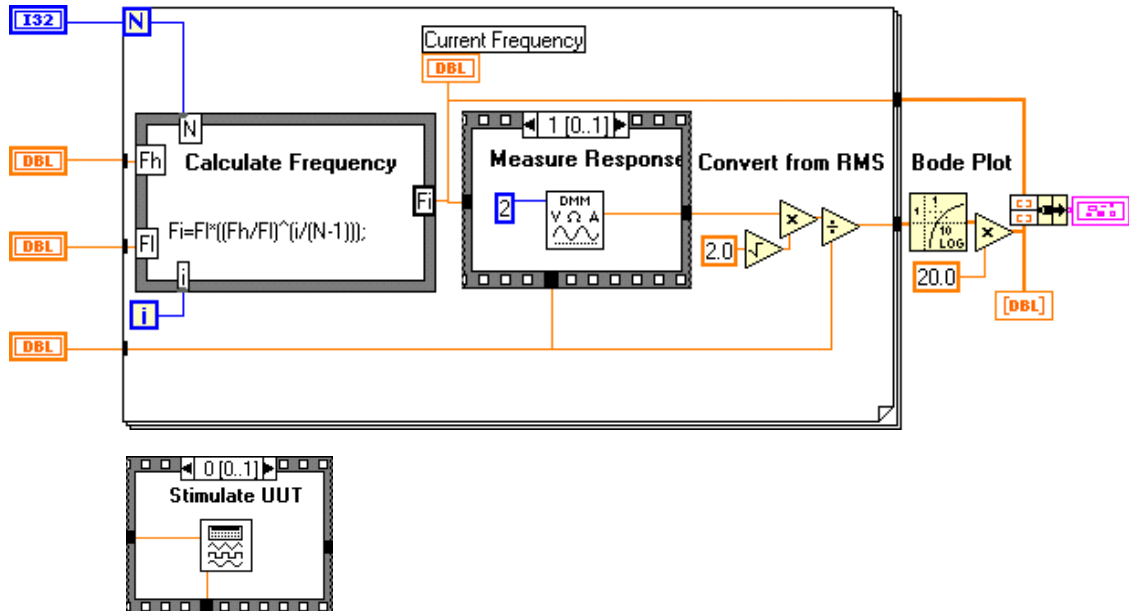


Fig. 7.2 Diagrama del VI Frequency response.

Cuestiones opcionales

1. ¿Cuál es la atenuación máxima del filtro que se puede medir con un error del 10%?
2. ¿Cómo afectaría el promediado de N medidas sobre los errores en la medida de amplitud y fase?
3. Una manera de reducir los errores debidos al ruido es realizando una detección coherente de la señal a la salida del filtro. En este caso se puede realizar esta detección coherente por software. Para compensar los errores de disparo del osciloscopio la detección coherente debe realizarse en ambos canales y luego hallar la diferencia de fases entre los dos canales. El módulo de la función de transferencia también se estima a partir del cociente de amplitudes obtenidas a la salida del detector coherente de ambos canales. El diagrama de bloques, para el método de medida propuesto, sería el de la figura 7.3. El tiempo de integración deberá ser un número entero de ciclos de señal. Realizar un subVI que permita estimar el módulo y la fase de la función de transferencia a partir de dos vectores de entrada que contengan la señal adquirida a la entrada y la salida del filtro.

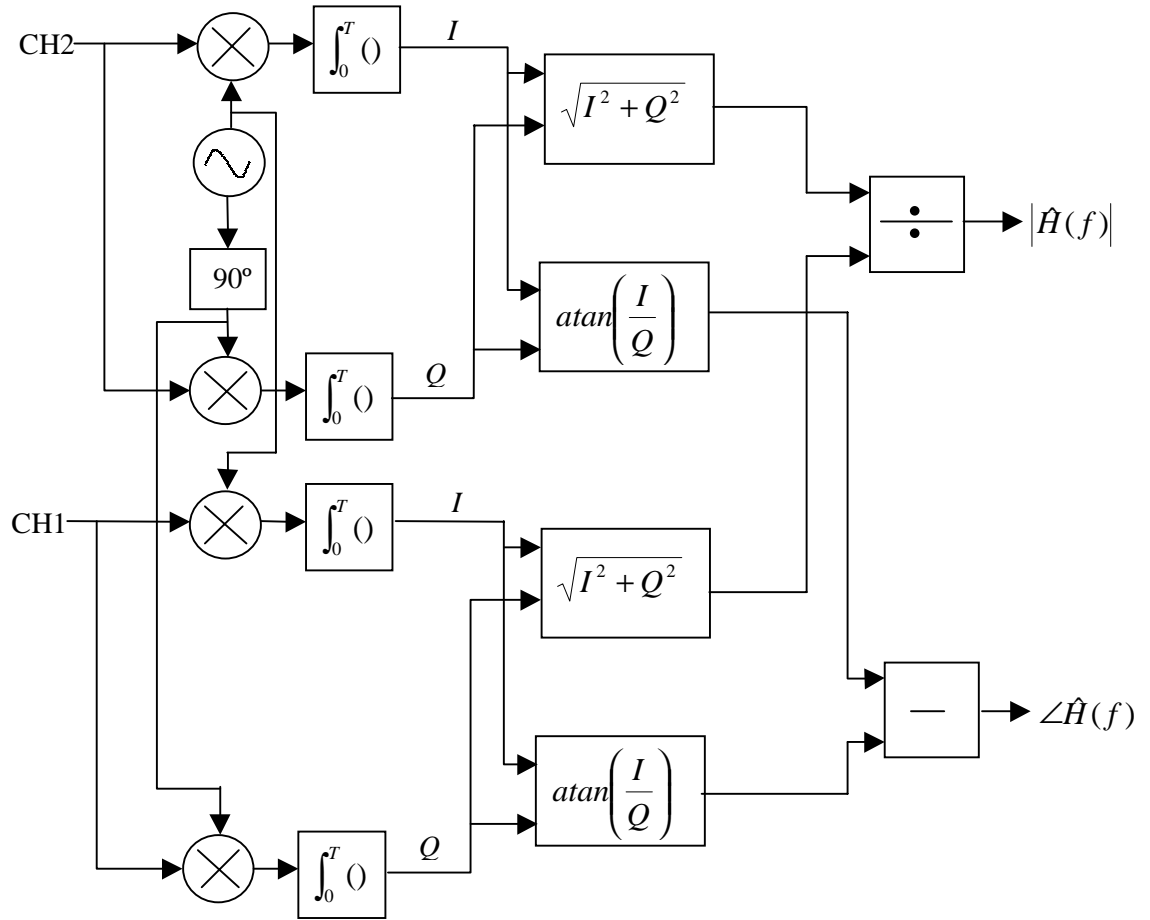


Fig. 7.3 Diagrama de bloques del VI para medir la función de transferencia del filtro utilizando demodulación coherente.

NOTA: Los componentes y la placa de pruebas (así como los cables) no serán suministrados por el laboratorio.

Práctica 8-a

Medida de las características de un amplificador operacional (AO)

Objetivos

El objetivo de esta práctica es introducir al alumno en el diseño de sistemas y equipos de prueba automáticos (ATE) de tipo bastidor. Para ello se diseñará un sistema que permita obtener de forma automática algunas de las características de un amplificador operacional. Para su realización se dispone de instrumentos controlados mediante un bus IEEE-488, de una placa de adquisición con E/S digital conectada al BUS del PC y de una placa de circuito impreso diseñada *ex profeso* para esta aplicación. El conjunto se muestra en la figura 8a.1

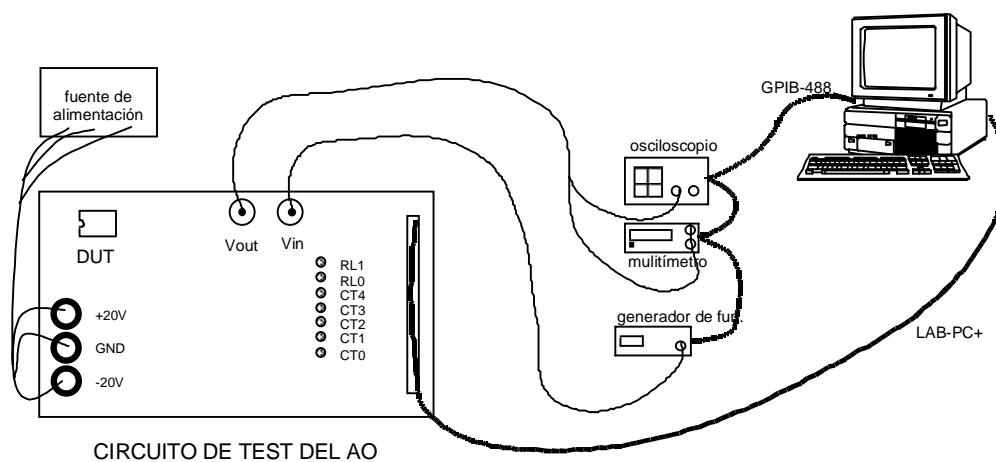


Fig. 8a.1 esquema de bloques del sistema para medida de los parámetros de un AO

Parámetros a determinar

Los parámetros que se deberán obtener del dispositivo bajo prueba son:

- Tensión de desequilibrio (offset)
- Corrientes de polarización a la entrada (I_{b+} , I_{b-})
- Rechazo frente a las variaciones de la alimentación, tanto para la tensión positiva como para la negativa (PSRR+, PSRR-)
- Rechazo a la tensión de modo común a la entrada (CMRR)
- Ganancia en lazo abierto, en función de la frecuencia

El circuito disponible en el laboratorio para la realización de estas medidas está basado en circuitos ATE reales utilizados por los fabricantes de AO's, si bien se han simplificado algunas funciones y se ha reducido la complejidad de los circuitos. En la figura 8a.2 puede verse el esquema simplificado del este circuito.

Esta realización se basa en la inclusión de un amplificador de instrumentación auxiliar (A1) en el lazo de realimentación del AO que se quiere medir. La operación de este circuito está gobernada por tres interruptores S1, S2 y S3, dos fuentes de alimentación controlables V_{AO+} y V_{AO-} y una tensión de entrada V_{in} . La tensión de salida es V_{out} .

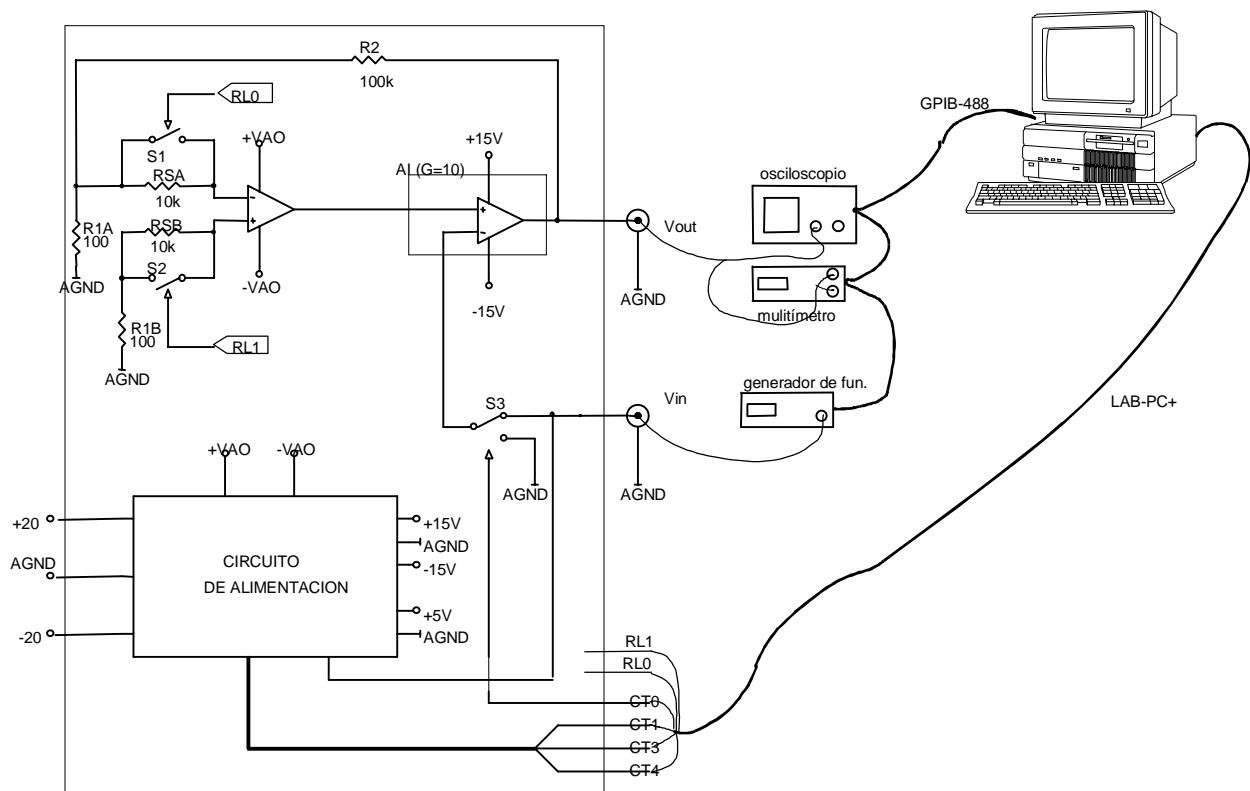


Fig. 8a.2 Esquema simplificado del circuito utilizado para la caracterización de un AO.

Los dos interruptores S1 y S2 se han realizado a base de microreles (reles REED) para reducir al máximo las corrientes de fugas a la entrada del AO. Estos interruptores permiten conectar dos resistencias R_{SA} y R_{SB} a las entradas del operacional para poder medir la caída de tensión que se produce al pasar por ellas la corriente de polarización de la entrada del AO. Las señales de control para activar los relés son RL0 y RL1. Estas señales están conectadas a los bits PA5 y

PA6 del puerto de E/S digital PA de la placa de adquisición LabPC+, que se encuentra en el bus del PC.

El conmutador S3 se ha realizado mediante interruptores analógicos basados en tecnología JFET (U2 en el esquema del anexo). S3 permite conectar la señal que proviene del generador de funciones (V_{in}) para realizar las medidas de ganancia en lazo abierto y CMRR. Se controla mediante la señal CT0 que está conectada al bit PA0 del puerto PA.

Las tensiones de alimentación del operacional a medir, V_{AO+} y V_{AO-} , se pueden hacer variar en función del valor de las señales CT1, CT3, CT4, V_{ref} y V_{in} . Esto permitirá realizar las medidas de PSRR y también de CMRR. Las expresiones que relacionan el valor de V_{AO+} y V_{AO-} con las señales anteriores son:

$$\begin{aligned} V_{AO+} &= V_{ref} - V_{in} * (1 - CT1) \\ V_{AO-} &= -V_{ref} - [V_{in} * (2 * CT4 - 1) * CT3] \\ &\text{(CT1, CT3 i CT4 corresponden al valor lógico de la señal, "1" o "0")} \end{aligned}$$

V_{ref} es una tensión de referencia incluida en el circuito. Las señales CT1, CT3 y CT4 se corresponden con los bits PA1, PA3 y PA4 del puerto PA.

Las posiciones de los interruptores que se muestran en la figura 8a.2 corresponden al nivel de reposo, es decir con la señal de control a nivel lógico 0.

La placa dispone de indicadores luminosos, LED's, que identifican el estado de cada una de las líneas de control.

La interfase entre las líneas del puerto PA de la placa LabPC+ y el circuito de medida se ha realizado mediante amplificadores inversores, para evitar cargar excesivamente las líneas de la placa de adquisición y protegerlas frente a eventuales cortocircuitos del sistema bajo medida. A la hora de realizar el programa de control deberá tenerse en cuenta esta inversión de signo.

Todos los niveles lógicos indicados en este manual corresponden a niveles lógicos en la placa

Control de las líneas de E/S mediante LabVIEW

LabVIEW dispone de una potente herramienta para diseñar aplicaciones en que intervengan tarjetas de adquisición con E/S analógicas o digitales: *DAQ Solution Wizard*. Se puede acceder a esta herramienta desde la pantalla inicial que aparece al arrancar LabVIEW. No obstante, los requerimientos de control de E/S en el diseño que se propone son lo suficientemente simples para que no sea necesario acudir a esta herramienta. Es suficiente con utilizar las funciones DIGITAL I/O de la librería DATA ACQUISITION. En concreto, será suficiente con usar la función WRITE TO DIGITAL PORT, que puede verse en la figura 8a.3.

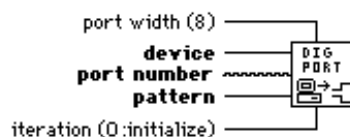


Fig. 8a.3 Función básica (de alto nivel) para escribir a un puerto digital de E/S

Esta función, de hecho es un VI realizado a base de unas funciones más elementales, llamadas ADVANCED DIGITAL I/O. En la figura 8a.4 puede verse la realización del VI WRITE TO DIGITAL PORT mediante funciones ADVANCED. Para el control requerido en la práctica se pueden usar cualquier conjunto de ellas.

De hecho, las funciones ADVANCED también son VI's, que llaman a un CODE INTERFACE NODE para poder ejecutar una rutina escrita y compilada en un lenguaje externo a LabVIEW.

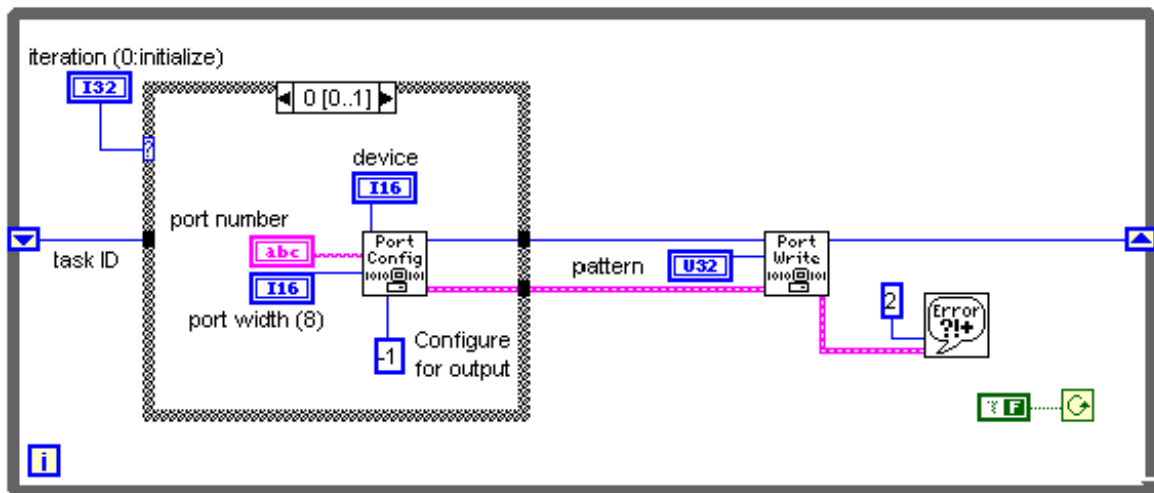


Fig. 8a.4 Realización de la función WRITE TO DIGITAL PORT mediante la funciones ADVANCED PORTCONFIG y PORTWRITE

Los parámetros necesarios para configurar y realizar escrituras en un puerto son el número de dispositivo y el del puerto. El número de dispositivo se asigna de forma automática al instalar y configurar las placas de adquisición. En nuestro caso solo hay una placa, por lo que tendrá asignado el número 1. El número del puerto depende del tipo de placa utilizada y del puerto utilizado (si la placa dispone de más de uno). En nuestro caso utilizamos el puerto PA de una tarjeta LabPC+, al que corresponde el número 0.

El siguiente diagrama muestra un VI que configura el puerto 0 del dispositivo 1 para que todas las líneas sean de salida, envía una palabra al puerto (pattern), realiza una lectura del mismo y muestra el valor de forma gráfica en una barra de LED's. Puede usarse esta realización para la práctica, pero el uso de la función WRITE TO DIGITAL PORT es más simple.

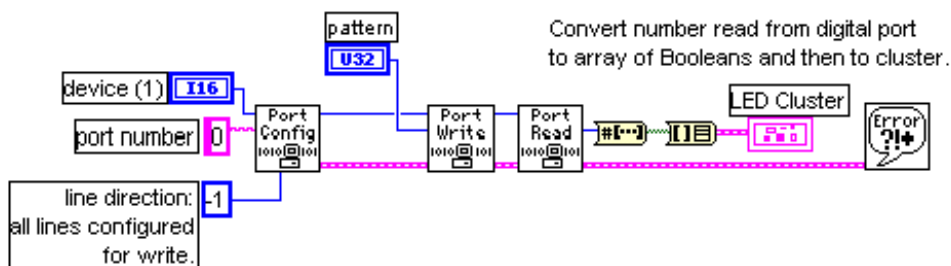


Fig. 8a.5 Ejemplo de configuración y salida de datos por un puerto E/S digital

Trabajo previo

Antes de proceder a la programación de los puertos de E/S y de los instrumentos es necesario determinar las expresiones que relacionan los parámetros a medir con el valor de las líneas de control y la señal de entrada.

Para la realización de las medidas de tensión de offset y corrientes de polarización no es necesario aplicar ninguna señal externa, y las tensiones de alimentación del AO se dejarán fijas al valor recomendado por el fabricante.

Análisis del circuito de medida para las características de entrada

Determinar las expresiones que relacionan la tensión de salida V_{out} con las corrientes de polarización y la tensión de offset del AO bajo prueba. Hacer el análisis para las distintas posiciones de los relés, con S3 conectado a masa. Suponer que el efecto de las corrientes de polarización sobre R_{1A} y R_{1B} es despreciable.

Determinación de las palabras de control

Cual es el valor de los bits del puerto PA para realizar cada una de las medidas necesarias. Completar la tabla siguiente.

Medida	RL0/PA5	RL1/PA6	CT0/PA0	CT1/PA1	CT2/PA2	CT3/PA3	CT4/PA4
V_{offset}							
I_{B+}							
I_{B-}							
los							

Análisis del circuito para la medida de características de transferencia

La medida de la ganancia en lazo abierto se realiza de forma indirecta mediante la inyección de una señal V_{in} en un punto dentro del lazo de realimentación. En este caso el punto escogido es el interruptor S3.

Determinar la expresión que relaciona la ganancia de lazo abierto del AO bajo prueba con la tensión V_{in} , la tensión de salida V_{out} , la ganancia del amplificador de instrumentación y las resistencias R_{1A} y R_{1B} .

Análisis del circuito para las medidas del rechazo a perturbaciones

Para las medidas de $PSRR+$ y $PSRR-$ deberemos variar cada una de las tensiones de alimentación del AO, V_{AO+} y V_{AO-} y ver el efecto en la salida. La medida se realizará por separado para cada una de las tensiones de alimentación. S3 debe estar conectado a masa y la señal senoidal que provoca la variación deberá estar conectada a V_{in} . El $PSRR$ viene dado por la expresión:

$$PSRR \approx 20 \log \left[\left(\frac{R_2}{R_{1A}} \right) \frac{V_{AO_{MAX}} - V_{AO_{MIN}}}{V_{OUT}} \right]$$

La medida del rechazo en modo común se debería hacer inyectando la misma señal a la entrada inversora y no inversora del AO. No obstante se ha escogido un método alternativo para realizar esta medida: si se hacen variar las dos alimentaciones del AO con la misma señal de control ($V_{in}=V_{AC}$) y en el mismo sentido, y simultáneamente se aplica esta misma señal a S3, el resultado es el mismo que aplicar la señal de control a las dos entradas del AO. La expresión del CMRR medido de esta forma es:

$$CMRR \approx 20 \text{LOG} \left[\left(\frac{R_2}{R_{1A}} \right) \frac{V_{AC}}{V_{OUT}} \right]$$

¿Cuál ha de ser el valor de los bits del puerto PA para las medidas de PSRR+, PSRR-, CMRR y la ganancia en lazo abierto?. Completar la tabla siguiente

Medida	RL0/PA5	RL1/PA6	CT0/PA0	CT1/PA1	CT2/PA2	CT3/PA3	CT4/PA4
PSSR+							
PSSR-							
CMRR							
A _o (s)							

Trabajo experimental

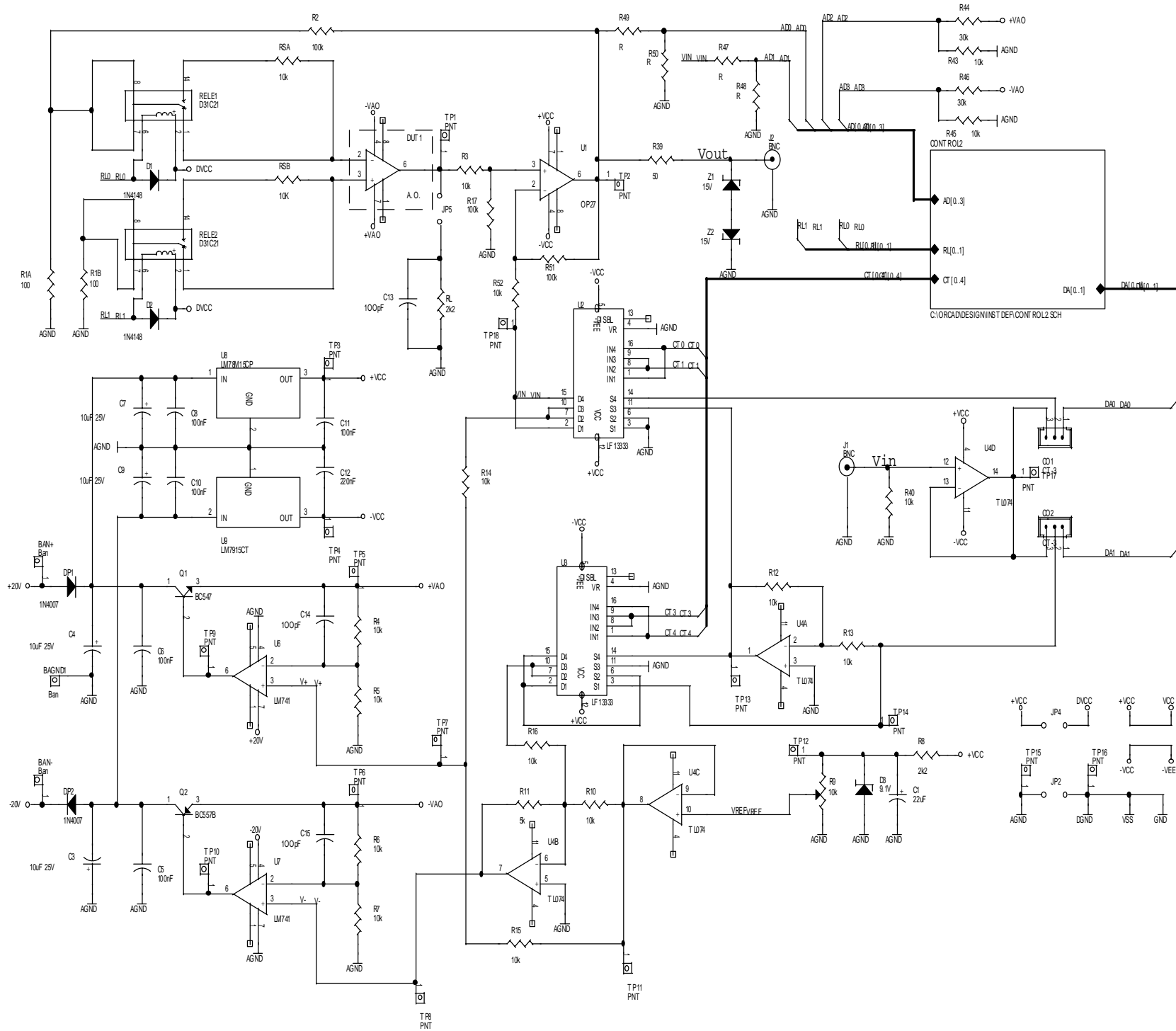
El operacional sobre el que realizar las medidas puede ser escogido por el alumno. Como indicación sugerimos usar un AO "clásico": $\mu A741$ y otro con entrada FET, también "clásico": TL071. Las características de estos dispositivos se pueden encontrar fácilmente en los manuales de los fabricantes (p.ej. Texas Instruments) y permitirán comprobar la fidelidad de las medidas realizadas. Si se elige otro dispositivo se deberá comprobar que el patillaje es compatible con el zócalo de la placa de prueba previsto a este efecto.

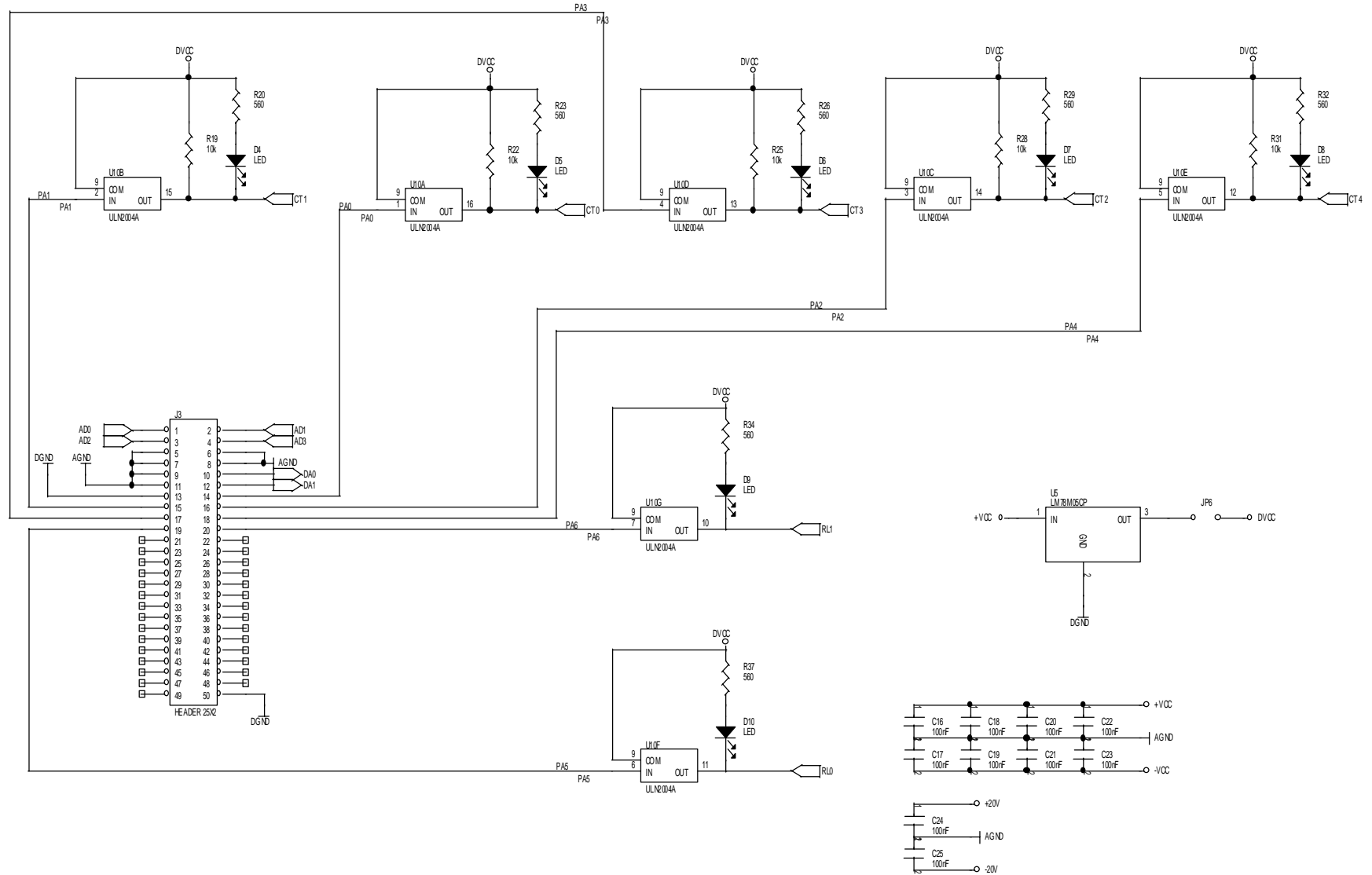
Diseñar un VI que realice las medidas de corrientes de polarización y tensión de offset.

Modificar el diseño anterior para realizar las medidas de PSRR, CMRR y AO a tres frecuencias distintas: 10 Hz, 1 kHz y 10 kHz en módulo y fase.

Diseñar un sistema completo de prueba que permita comparar las medidas con las especificaciones del fabricante y de una señal de alarma si el dispositivo esta fuera de tolerancia.

Antes de conectar las señales de control y excitación al circuito de prueba, comprobar que el programa funciona correctamente y que los valores de tensión son los correctos.





Práctica 8-b

Test de un VCO y un PLL

Objetivos

El objetivo de esta práctica es el obtener, de forma experimental, algunas de las especificaciones más esenciales en el diseño basado en osciladores controlados por tensión (VCO en inglés). La aplicación fundamental de los VCOs es en circuitos de conversión tensión-frecuencia, o en otras palabras en moduladores de FM. En los circuitos demoduladores de FM por lazo enganchado en fase (PLL) el VCO es también uno de los elementos fundamentales conjuntamente con el detector de fase y el filtro. En esta práctica utilizaremos un dispositivo comercial que incorpora un VCO y dos detectores de fase, este circuito es el HEF4046B. La frecuencia máxima generada por este circuito es de 1 MHz, lo que limita su aplicación en comunicaciones (su uso principal es en telemetría). Esta limitación, sin embargo, no nos afecta ya que con la instrumentación disponible sólo podemos medir con exactitud frecuencias hasta 300 kHz, por lo que el diseño se realizara teniendo en cuenta este último límite. Queremos resaltar, sin embargo, que al trabajar a estas frecuencias no quita generalidad a los conceptos que se tratan ni a los métodos experimentales utilizados.

Descripción del sistema de medida a utilizar

Para realizar las medidas se dispondrá de los siguientes instrumentos controlables: multímetro (con posibilidad de medir frecuencia hasta los 300 kHz), osciloscopio digital, generador de funciones y fuente de alimentación. Además se dispone de una placa de adquisición de datos conectada al bus del PC con entradas/salidas analógicas y digitales. Las salidas del conversor D/A se han configurado en modo unipolar con un fondo de escala de +10 V.

La placa de adquisición del PC (LAB PC+) dispone de dos convertidores Digital Analógico (DA), un conversor Analógico Digital (AD) y tres puertos (PA, PB y PC) de entrada o salida digitales de ocho bits cada uno.

Las librerías de "data acquisition" de LabVIEW reconocen esta placa de adquisición y ofrecen distintos niveles de programación, desde los controladores más sencillos de uso (Easy) a las funciones

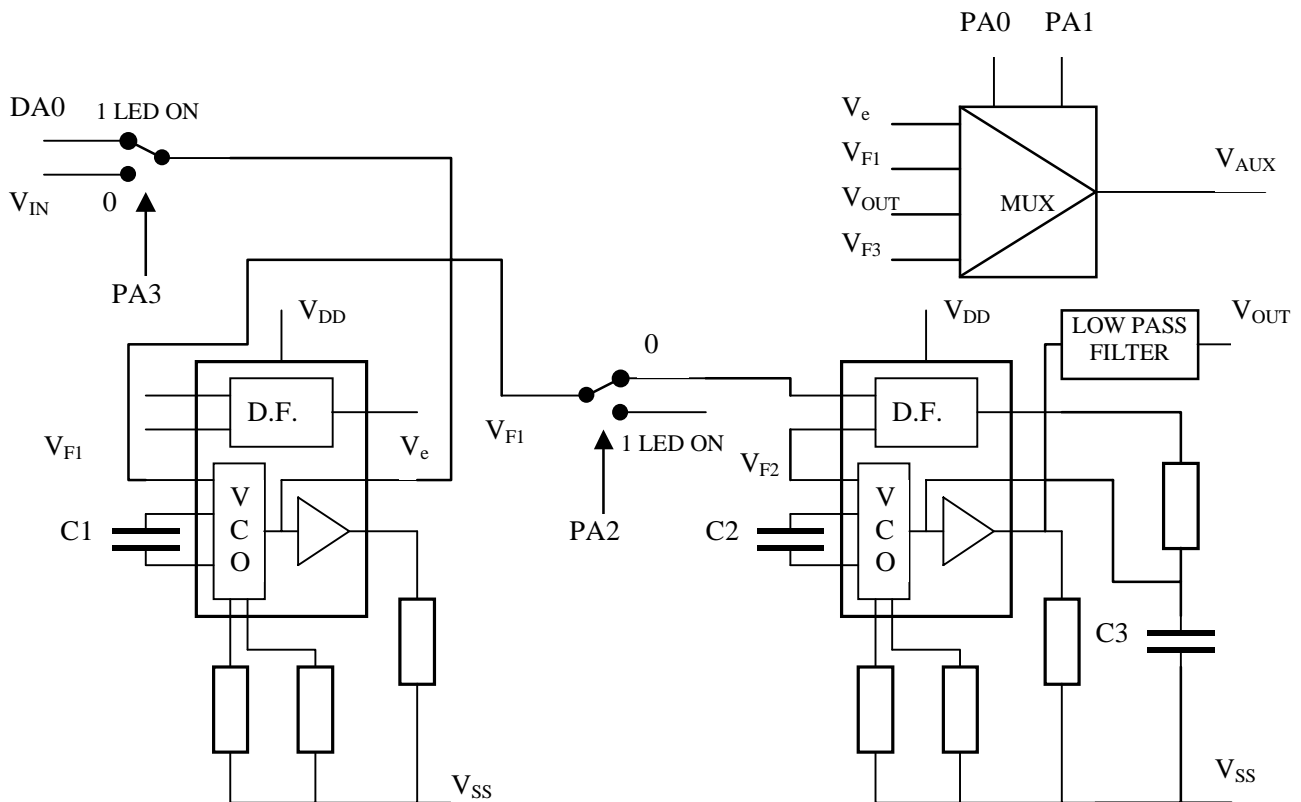


Fig. 8b.1. Circuito modulador y demodulador basado en VCO y PLL

más especializadas. Para estas prácticas es suficiente con la funcionalidad que ofrecen los controladores más sencillos.

Como sólo se dispone de una placa de adquisición los controladores la identifican como el DEVICE

1. Los puertos PA, PB y PC se identifican como los CHANNELS 0, 1 y 2 respectivamente

Descripción del circuito a caracterizar

El circuito diseñado es un modulador-demodulador de FM utilizando dos circuitos HEF4046B (figura 8b.1). El primero de ellos es utilizado como VCO. Su alimentación es de +10 V ($V_{DD}=+10$ V, $V_{SS}=0$ V) con una excursión de frecuencia diseñada para ir desde 100 kHz a 200 kHz (figura 2).

El segundo HEF4046 constituye el demodulador de FM basado en un lazo enganchado en fase (PLL) que se utiliza para recuperar la señal de entrada V_e . Su alimentación también es de +10 V y los valores del circuito se han tomado para conseguir las siguientes especificaciones:

- frecuencia central: $f_0 = 150$ kHz
- margen de seguimiento: $2f_L = 100$ kHz
- margen de captura: $2f_c = 45$ kHz
- ancho de banda (-3 dB) del filtro de 2º orden: 1 kHz

Las señales accesibles para realizar el test son:

V_{DD} Alimentación de +10 V

- V_e señal de entrada al VCO (voltaje limitado entre 0 y +10 V). Se puede escoger entre dos opciones: V_{IN} (entrada con coaxial) o DA0 (entrada conectada al convertor digital/analógico de la placa del PC).
- V_{F1} señal de salida del VCO1 (es una señal cuadrada entre 0 y +10 V, a la frecuencia fijada por el VCO en función de V_e)
- V_{F2} señal de salida del VCO2 (es una señal cuadrada entre 0 y +10 V, a la frecuencia fijada por el VCO2 en función la fase detectada en el PLL)
- V_{OUT} Salida del PLL filtrada paso bajo
- V_{AUX} Salida auxiliar que se puede conectar, gracias a un multiplexor, a V_e , V_{F1} , V_{OUT} ó V_{CO2}
- PA0 y PA1 Controles del multiplexor
- PA2 Desconexión del emisor

Tabla 8b.1 Valores de las señales de control para la selección de la señal V_{AUX}

PA0	PA1	VAUX
0	0	V_e
0	1	V_{F1}
1	0	V_{OUT}
1	1	V_{F2}

ATENCIÓN: Tensiones en las entradas mayores que la alimentación positiva o menores que la negativa dañan **irreversiblemente** los circuitos CMOS.

Especificaciones del VCO y del demodulador

Las características del VCO son las que determinan en mayor grado las especificaciones del sistema completo. Entre otras, las más importantes son:

- margen de variación de la frecuencia en función de V_e
- linealidad del VCO
- estabilidad en temperatura de la frecuencia

El margen de variación teórico y la linealidad del VCO esta definida por el fabricante tal como se representa en la figura 8b.2.

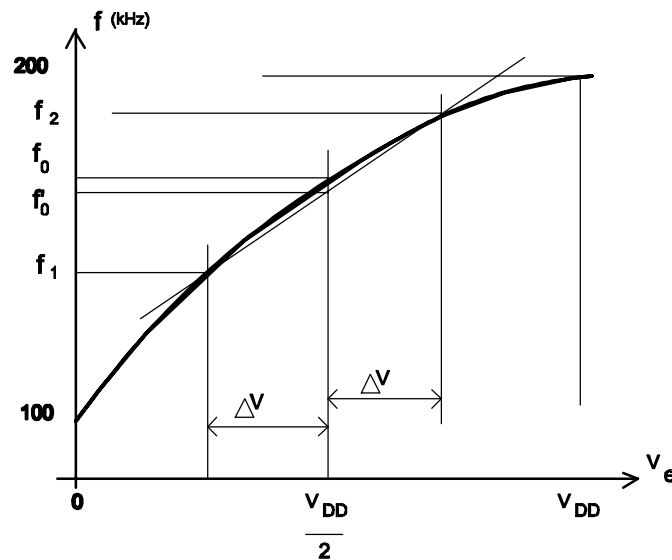


Fig. 8b.2 Error de linealidad de un VCO

$$f_{o'} = \frac{f_1 + f_2}{2}$$

$$Er. no lin. = \epsilon_L = \frac{f_{o'} - f_0}{f_{o'}} 100 \quad [\%]$$

$$\Delta V = 2.5V \quad para \quad V_{DD} = 10V$$

La estabilidad térmica se da como la variación relativa de la frecuencia de salida ($V_e = V_{DD}/2$) dividido por grado centígrado.

$$Coef. Temp = \frac{f(T_0) - f(T_0 + \Delta T)}{f(T_0)} \frac{100}{\Delta T_0} \quad [\% / ^\circ C]$$

En el Demodulador basado en PLL los parámetros de diseño más importantes son:

- margen de enganche
- margen de seguimiento

El margen de enganche se define como el intervalo frecuencial en que el PLL se engancha en fase con la señal de entrada cuando previamente estaban desenganchados.

El margen de seguimiento se define como el intervalo frecuencial en que el PLL puede seguir la frecuencia de entrada si está inicialmente enganchado en fase.

Visto el sistema modulador-demodulador como un canal de transmisión de una señal analógica la calidad del sistema se especifica, entre otros, por los siguientes parámetros:

- respuesta en frecuencia (ganancia y fase)
- margen dinámico de la señal
- distorsión armónica total (ó linealidad)
- respuesta en continua: offset y su estabilidad
- ruido del sistema

Tener en cuenta que el margen dinámico de la señal dependerá del margen dinámico de los circuitos diseñados con el HEF4046 pero también del operacional del filtro paso bajo de la salida.

Métodos de medidas de las especificaciones del VCO

Para medir la relación frecuencia de salida-tensión de entrada se necesita por un lado aplicar una tensión continua (conocida con suficiente exactitud) y medir, igualmente con suficiente exactitud, la frecuencia de salida. Es aconsejable generar la tensión continua con el convertor D/A de la placa de PC porque permite modificar su amplitud fácilmente para hacer el barrido en tensión. A partir del código digital enviado al convertor D/A podemos estimar la tensión aplicada al circuito. Sin embargo, para estimar esta tensión el instrumento más exacto disponible es el multímetro. Por lo que puede ser conveniente, para reducir los errores, el medir con este instrumento. La frecuencia de salida puede medirse con el multímetro en su función de frecuencímetro. Lo ideal sería disponer de dos instrumentos: multímetro y frecuencímetro, pero se puede realizar la medida secuencialmente conmutando las señales con el multiplexor incorporado en la placa.

Una vez obtenida la curva anterior se puede deducir de ella la linealidad del VCO aplicando la definición dada por el fabricante.

El método de medida será pues:

INSTRUMENTOS

- I0: Fuente de alimentación
- I1: Placa LAB-PC+ (1 convertor D/A y 4 líneas de salida digitales)
- I2: Multímetro (función de medida en DC y en frecuencia)

CONEXIONES

- La salida del D/A la conectaremos a V_e .

NOTA: Cuando se desengancha el PLL tiende a oscilar a su frecuencia libre. En el diseño realizado esta frecuencia es la central (equivalente a una entrada de 5 V).

- Las líneas PA0 y PA1 se corresponden a las líneas de menos peso del puerto PA.
- Entrada del multímetro conectada a V_{AUX} .
- (el osciloscopio puede estar conectado para supervisar la medida a la entrada V_e (CH1) y a la salida V_{F1} (CH2))

PROTOCOLO

- 1- Iniciar la medida a fondo de escala del convertor D/A
- 2- Conectar V_{AUX} a V_e

- 3- Medir la tensión
- 4- Conectar V_{AUX} a V_{F1}
- 5- Medir la frecuencia
- 6- Reprogramar el conversor D/A (decrementos de 0.2 V)
- 7- Volver a 2 hasta que $V_e=0$

Para la determinación del coeficiente de temperatura deberíamos fijar la tensión de entrada a un valor en el centro del margen y medir el cambio de la frecuencia sometiendo al dispositivo a un incremento de temperatura conocido. Esta medida no se realizará.

Métodos de medida de las especificaciones del conjunto

El margen de enganche se puede medir realizando los siguientes pasos:

- desenganchar el PLL (por ejemplo desconectando la señal de entrada al comparador de fase, mediante la señal de control PA2)
- hacer oscilar el VCO a su mínima frecuencia (ó a la máxima para el otro extremo)
- conectar la señal
- modificar la frecuencia del VCO hasta detectar el enganche de fase

Para medir el margen de seguimiento basta con modificar la frecuencia hasta detectar el desenganche de fase.

NOTA: Recordar que la tensión de entrada solo puede variar entre 0 y V_{DD} . Por lo tanto hay que sumar una tensión continua a la señal del GF para no tener valores de tensión negativos. Se recomienda verificar esta condición antes de conectar la señal al circuito.

Para medir la respuesta en frecuencia lo más apropiado es atacar el circuito con una señal senoidal utilizando el Generador de funciones (GF) y realizar un barrido en frecuencia. El osciloscopio nos permitirá realizar las medidas de relaciones de módulo y fase entre la entrada y la salida si las adquirimos simultáneamente. Con la placa de adquisición del PC también se podría realizar esta medida pero la resolución en tiempo sería menor.

La distorsión armónica total se puede medir inyectando una senoide conocida con el GF y midiendo el incremento de la potencia de los armónicos a la salida del sistema con el osciloscopio, para ello puede adquirirse la señal temporal con el osciloscopio y utilizar las librerías de análisis de señal para hacer la estimación del espectro y el cálculo de la distorsión armónica total. Seleccionar un enventanado adecuado de la señal para realizar la FFT. Serán más apropiadas las ventanas que den menos resolución en frecuencia y más en amplitud.

Trabajo previo

1. Definir para cada una de las medidas que se proponen a continuación que equipos del laboratorio se utilizarán, como se interconectarán y el protocolo para realizar las medidas:
 - 1.1. Margen de enganche y seguimiento
 - 1.2. Respuesta en frecuencia (función de transferencia)
 - 1.3. Distorsión armónica total (con una señal de entrada de 5 V pico a pico de amplitud y 200 Hz)
2. Explicar que método hay que seguir para calcular la distorsión armónica total a partir de la amplitud de los armónicos de 200 Hz que tenemos dentro de la banda de paso de nuestro sistema (1 kHz).

Trabajo de laboratorio

1. Realizar los programas (VI) para automatizar cada una de las siguientes medidas:
 - A - margen de variación de la frecuencia en función de V_e
 - B - linealidad del VCO
 - C - margen de enganche y seguimiento
 - D - respuesta en frecuencia (función de transferencia) (con una señal de entrada de 2.5 Vpp de amplitud centrada en 5 Voltios)
 - E - distorsión armónica total (con una señal de entrada de 5 Vpp de amplitud y 200 Hz)
2. Realizar las medidas y comparar las especificaciones obtenidas con las del fabricante, en el caso que éste las suministre, o con las previstas por diseño.
3. Rediseñar los VI que se han realizado para que puedan funcionar como subVI en una aplicación de test completa. Se recomienda añadir una salida booleana a cada subVI (PASA/NO PASA) y que el programa ejecute secuencialmente los test hasta que uno no pase o se complete el test de todos los parámetros correctamente (en los ejemplos de LabVIEW puede encontrarse una aplicación de este tipo).

Práctica 9

Control de Instrumentos a través de Internet

Objetivos

El objetivo de esta práctica es dar una idea general al alumno de las posibilidades del control de instrumentación remota a través de Internet. Para ello, se propone el estudio del estado de calibración de los multímetros del laboratorio. Todos los multímetros del laboratorio se conectarán a una única fuente de tensión continua y los alumnos deberán crear una aplicación cliente y otra servidor que permita adquirir las medidas de los multímetros de los diferentes puestos. Se les propone que utilicen alguna de las técnicas de análisis estudiadas en el tema de teoría de errores para la detección de errores sistemáticos (test de medias o análisis de la variancia).

Introducción a las funciones TCP/IP de LabView

Transferencia de datos mediante TCP

El protocolo TCP asegura una transmisión fiable a través de la red, devolviendo la información en la secuencia correcta y sin errores, pérdidas o duplicación. Cuando se envían datos mediante el protocolo TCP, se le añade información adicional y se transfiere al protocolo IP, que fragmenta los datos en paquetes, también conocidos como datagramas, y los transmite. En la recepción el proceso se invierte y el protocolo TCP devuelve los datos libres de errores y en el orden correcto.

Inicio de la conexión

La comunicación se puede iniciar mediante la espera de una nueva conexión o bien buscando de forma activa una conexión con una dirección específica. Al establecer una conexión TCP, se debe especificar la dirección y el puerto que se quiere asignar en esa dirección. Un puerto se representa mediante un número entero sin signo de 16 bits, entre 0 y 65535. Diferentes puertos en una dirección identifican diferentes servicios en esa dirección y simplifican el manejo de conexiones simultáneas múltiples.

Establecer una conexión de forma activa

Para establecer la conexión de forma activa desde una aplicación en una dirección y puerto específicos se puede utilizar la función *TCP Open Connection*. Donde los parámetros de entrada aparecen en la figura 9.1. Si la conexión ha tenido éxito, la función devuelve el identificador de conexión, *connection ID*, que es único para esa conexión. Se debe utilizar este identificador para la llamada a las demás funciones de la librería.

TCP Open Connection

Attempts to open a TCP connection with the specified address and port.

Click the parameters for more information.

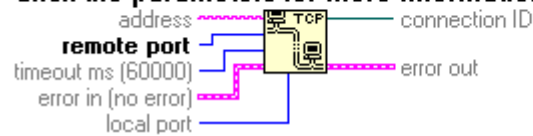


Figura 9.1 Función para crear una nueva conexión TCP.

Esperar a una conexión entrante

Se pueden utilizar en LabView dos métodos para esperar a una nueva conexión entrante:

- En el primer método, se utiliza el VI de la librería TCP, *TCP Listen*, para crear un receptor y esperar a una conexión válida en un puerto específico. Si la conexión ha tenido éxito, el VI devuelve un identificador de conexión, la dirección y el puerto de la conexión TCP remota.

TCP Listen

Creates a listener and waits for an accepted TCP connection at the specified port.

Click the parameters for more information.

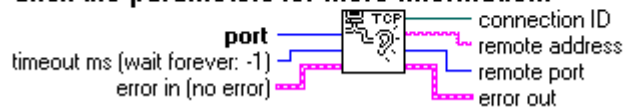


Figura 9.2 Función que crea un Listener y espera a una nueva conexión TCP remota.

- El segundo método consiste en utilizar la función *TCP Create Listener*, y a continuación utilizar la función *TCP Wait on Listener* para aceptar nuevas conexiones. La función *TCP Wait on Listener* devuelve el mismo identificador, *Listener ID in*. Cuando se ha terminado la espera para nuevas conexiones, se puede cerrar la conexión de espera con *TCP Close*. No se puede escribir o leer de un *Listener*.

TCP Create Listener

Creates a listener for a TCP connection.

Click the parameters for more information.

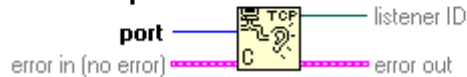


Figura 9.3 Función para crear un Listener sin esperar conexión.

TCP Wait on Listener

Waits for an accepted TCP connection at the specified port.

Click the parameters for more information.

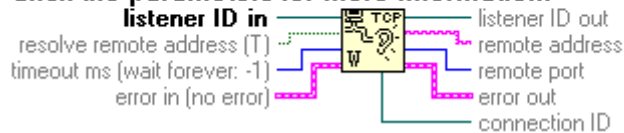


Figura 9.4 Función para esperar a una nueva conexión en un Listener ya creado previamente.

La ventaja de utilizar el segundo método estriba en que se puede cancelar la operación de escucha llamando a *TCP Close*. Este método es útil para aplicaciones en las cuales se quiere recibir de una conexión sin utilizar un *timeout*, pero se quiere cancelar la espera cuando se cumpla otra condición (por ejemplo, cuando el usuario pulsa un botón).

Lectura y escritura de datos

Una vez establecida la conexión, se pueden enviar y recibir datos con la aplicación remota utilizando las funciones *TCP Read* y *TCP Write*.

TCP Read

Receives a maximum of **bytes to read** bytes from the specified TCP connection, returning the results in **data out**.

Click the parameters for more information.

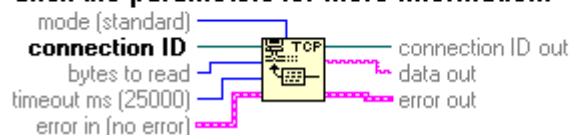


Figura 9.5 Función para leer los datos recibidos en un puerto TCP.

TCP Write

Writes the string **data in** to the specified TCP connection.

Click the parameters for more information.

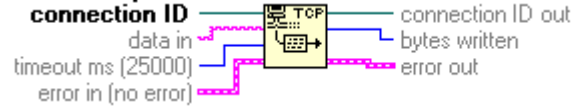


Figura 9.6 Función para enviar datos a un puerto TCP.

Los datos a enviar sólo pueden ser de tipo string. Por tanto, se deben convertir previamente a formato string para poder ser enviados a través del canal TCP/IP. Hay una función dentro de grupo de *advanced/data manipulation* que se llama *Type/Cast* que permite realizar la conversión en los dos sentidos tal como se muestra en la figura 9.1.

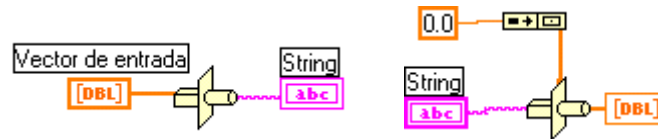


Fig. 9.7 Conversión de del formato double a una secuencia de bytes (tipo string) independiente de la máquina, para poder ser incluida dentro de un datagrama.

El tamaño del string no se conoce a priori, depende del tipo datos a enviar y debe establecerse algún medio para poder recuperar los datos originales. Una solución puede ser formatear los datos en bloques de tamaño fijo, por ejemplo si se trata de una sola medida. O bien, hacer dos envíos a través del canal TCP/IP. En el primero se envía el tamaño del string de datos en un string de longitud fija, 4 bytes, y en el siguiente mensaje irá el string de datos.

Sistema de medida

Uno de los objetivos de esta práctica es intentar identificar posibles errores sistemáticos en alguno de los multímetros de laboratorio realizando medidas sobre una tensión conocida. El procedimiento habitual, sería disponer de una referencia de tensión calibrada con una incertidumbre inferior a la del equipo a calibrar, y realizar un conjunto de medidas suficiente para poder reducir los errores aleatorios y determinar si existe un error sistemático. Como no es fácil ni barato disponer de una referencia de tensión de este tipo se ha decidido realizar medidas simultáneamente con más de un multímetro sobre una tensión suficientemente estable durante el tiempo de medida. La medida no permitirá hacer una "calibración absoluta" de los multímetros, pero al menos permitirá identificar aquellos en los que el resultado de las medidas se aleje en media más de lo esperado del resto. Para la referencia de tensión se ha utilizado una integrada bajo ruido y bajo consumo, 2,5 V (LM385). Para reducir más el ruido se ha alimentado con una pila de 9 V.

Cuando las medidas se realizan con dos multímetros, la comparación se puede realizar aplicando un test de medias, si las lecturas no se han realizado simultáneamente, o bien un test de lecturas apareadas. Si se han hecho medidas con más de dos multímetros, el procedimiento habitual es

realizar un análisis de la variancia para determinar si la diferencia entre las medias se debe a los errores aleatorios, o bien existen errores sistemáticos en algunos de los equipos. Para llevar a cabo esto, el LabView dispone una librería de análisis estadístico para realizar el análisis de variancias, *ID ANOVA.vi*

Trabajo previo

Determinar en cada puesto de trabajo la dirección IP del ordenador.

Determinar el factor confianza para el test de medias y de lecturas apareadas con un nivel de confianza del 95%.

Trabajo de laboratorio

Para el control remoto del multímetro a través de la red se utilizará el ejemplo *tcpex.llb* que contienen los VI *Data Server.vi* y *Data Client.vi* que permiten enviar datos entre la aplicación *Server* y *Client*. La dirección IP del servidor se especifica en la variable *localhost* y el puerto asignado a esa conexión en la variable *port*. El valor del puerto debe ser el mismo en la aplicación cliente y servidor, no utilizar valores de puerto inferiores a 3000.

9.1 Modificar la aplicación servidor para que inicialice el multímetro en medida de tensión continua con 5½ dígitos, y envíe el vector de medidas.

9.2 Modificar la aplicación cliente para que permita la conexión con más de un servidor.

9.3 Una vez leídos los datos de los diferentes equipos se tratará de identificar los posibles errores sistemáticos presentes en las medidas. La comparación puede hacerse de diferentes maneras:

- Lecturas apareadas (solo es posible si se han hecho simultáneamente las medidas con dos equipos, lo cual es difícil de garantizar)
- Comparación de medias, permite comparar las medias de las medidas realizadas con dos equipos.
- Análisis de la variancia (ANOVA). El LabView incorpora una librería de análisis que permite realizar el test para uno, dos y tres factores (1D, 2D, 3D)

9.4 En vista de los resultados obtenidos, ¿se debería calibrar algún equipo?

Trabajo de laboratorio opcional

9.5 Modificar las aplicaciones cliente y servidor para que se puedan enviar órdenes de programación del multímetro desde el cliente.

9.6 Repetir análisis del apartado 9.3 para medidas de resistencia. Utilizar la misma resistencia para todas las medidas.