

Conceptos esenciales sobre controladores y comandos nativos de los instrumentos¹ / 1ª parte

Artículo cedido por Agilent Technologies

Los controladores de instrumentos suelen ocupar el primer lugar en la lista de los peores quebraderos de cabeza para los ingenieros. En su intento por facilitar el control de los instrumentos, sus fabricantes y los grupos sectoriales han desarrollado distintos estándares para la conexión de instrumentos. Lo malo es que ello ha complicado y, en ocasiones, dificultado las opciones que han de elegir los ingenieros.

En este artículo, que consta de dos partes, intentaremos descifrar algunas de estas complejidades. En esta primera parte daremos respuesta a las preguntas más comunes sobre los comandos originarios de los instrumentos que se envían desde el PC al instrumento. Ofreceremos una breve explicación introductoria, describiremos cómo los distintos niveles o capas de software ayudan a comunicar el instrumento con el PC, y presentaremos los controladores de instrumentos y el papel que desempeñan en la comunicación entre un PC y los instrumentos. En la segunda, explicaremos los distintos tipos de controladores, mostrando una comparativa entre ellos, mostraremos cómo conseguir que el software y el hardware sean intercambiables, detallaremos la compatibilidad entre los controladores y el entorno de desarrollo elegido por el usuario, y ofreceremos algunas direcciones de uso común donde localizar controladores y herramientas asociadas a ellos.

Hewlett-Packard inventa HP-IB, que más adelante pasaría a ser GPIB

Antes de 1970, un formato muy popular para el control de dispositivos o instrumentos desde ordenadores era la E/S paralela, con la que se conectaba al instrumento un cable de 64 patillas. Cada una de las líneas del cable representaba una función o un rango de valores, y la cosa

consistía simplemente en poner a tierra la línea en cuestión en el instante adecuado. Un dispositivo, por ejemplo un voltímetro, devolvía un valor utilizando el formato decimal codificado en binario (BCD) 1-2-4-8, o incluso otro formato más exótico como 1-2-2-4. La sintaxis de programación no estaba estandarizada en absoluto. Sin embargo, al estar todo programado de forma fija e inalterable, el proceso era bastante sencillo e inmediato.

En 1971 se empezó a trabajar en el desarrollo de una interfaz de hardware estandarizada. La idea era poder disparar varios instrumentos a la vez y que, al mismo tiempo, fuera posible que hubiese otros instrumentos, tanto rápidos como lentos, "hablando" a través del mismo bus sin causar conflictos. Los primeros productos en utilizar este bus fueron lanzados por Hewlett-Packard (hoy Agilent Technologies) en 1972. El mismo año, este nuevo bus fue apodado "Hewlett-Packard Interface Bus" (HP-IB). En 1975, el IEEE lo adoptó como estándar sin apenas modificaciones, y nació el estándar IEEE-488. La interfaz original es conocida hoy popularmente como GPIB (General Purpose Interface Bus).

Software de entrada/salida (E/S)

Todo el software necesario para comunicarse con un periférico y controlarlo. Los comandos nativos del instrumento y la librería de funciones VISA son dos ejemplos de software de E/S que permite a los controladores y programas de aplicación acceder a la instrumentación de los periféricos.

Con la llegada de los PCs, surgió la necesidad de un software o lenguaje de comunicaciones de E/S común. La escasa potencia de procesamiento disponible entonces exigía una sintaxis sencilla, por lo que se eligieron comandos ASCII. He aquí un ejemplo para un voltímetro: "F1R2T1". Este comando significa "Seleccionar la función de medida de voltios de continua, la escala de 1 voltio y disparar una lectura."

Introducción de SCPI (Direct I/O) en 1989

En 1989, ante la necesidad de una claridad y facilidad de intercambio mayores que las que ofrecía el simple código ASCII, Hewlett-Packard introdujo un lenguaje de programación denominado TMSL (Test & Measurement Systems Language). En menos de un año, nueve fabricantes de equipos de medida y prueba se habían reunido para generar un método universal de control de instrumentos, utilizando TMSL como base. El resultado fue el estándar SCPI (Standard Commands for Programmable Instruments).

SCPI (Standard Commands for Programmable Instruments)

SCPI (pronunciado como "skippy") es un conjunto estándar de comandos nativos de instrumentos (también conocido como "direct I/O") que se utilizan para controlar dispositivos programables. Véase la dirección: <http://www.scpiconsortium.org>.

En la actualidad, SCPI sigue siendo el método de control de instrumentos más utilizado, por su rapidez de ejecución y porque abarca el 100% de las funciones programables del instrumento, por muy crípticas que sean. En SCPI, la sintaxis de programación del instrumento pasó a ser mucho más robusta y predecible. SCPI define una jerarquía estricta, y cada comando está asociado a una respuesta concreta. Estos comandos y respuestas se definieron para dispositivos de tipo fuente, medidor y conmutador. He aquí un ejemplo de código SCPI: CONF:VOLT:DC 0.3,0.003

Este comando indica al instrumento que se configure a sí mismo con el fin de prepararse para leer una señal de 0,3 voltios con una resolución de 3 milivoltios. Los comandos SCPI deben ser interpretados sintácticamente por el voltímetro (no todo voltímetro tiene una escala de 300 mV), lo cual añade al sistema un ligero nivel de retardo.

Sin embargo, para comunicarse con un ordenador hace falta más

software de E/S que los simples comandos del instrumento. Para poder enviar un comando al instrumento, es necesario definir la ruta de E/S que atraviesan los datos, seleccionar la tarjeta de E/S adecuada, averiguar en qué lugar del bus se encuentra el instrumento, y hablar con éste con la sintaxis correspondiente al tipo de E/S que se esté utilizando. Suponiendo que la tarjeta de E/S GPIB del ordenador se encuentra en la dirección 7 y que el multímetro digital está en la dirección 22 del bus, este podría ser un ejemplo de comando básico:

```
ASSIGN @Dvm to 722 !indica al ordenador
dónde enviar el comando
OUTPUT@Dvm; "TRIG:SOURCE:INT" !selecciona
como fuente de disparo la interna
```

Desarrollo del software de E/S SIDL para conseguir la independencia con respecto a la interfaz

SCPI funciona con una interfaz GPIB, pero si intentamos hacer lo mismo con el bus RS-232, la sintaxis será muy diferente. Cambiar entre GPIB y RS-232 obliga a reescribir mucho código. Y es ahí donde interviene el software de E/S SIDL (Standard Instrument Control Library).

SIDL (Standard Instrument Control Library)

SIDL es una librería de llamadas a funciones de E/S implementada y soportada principalmente por Agilent. Algunas de estas funciones básicas son comunes para todas las interfaces físicas (GPIB, RS-232, etc.), mientras que otras son específicas de cada interfaz. La librería SIDL proporciona un control muy completo y flexible de los instrumentos. Esta API viene incluida en la familia de librerías de E/S Agilent IO Library Suite 14.0. SIDL proporciona un acceso total a los productos GPIB, RS-232, LAN, y a los VXI basados en mensajes y en registros.

SIDL fue desarrollado por HP (Agilent) para independizar todo lo posible el software con respecto del sistema de E/S. SIDL añade una capa sobre el código del instrumento. Dicha capa comprueba qué tipo de E/S se está utilizando y altera la sintaxis en consecuencia. SIDL está optimizado para su uso desde

programas en C o C++, pero también puede utilizarse desde Visual Basic y otros entornos capaces de llamar a DLL arbitrarias de Windows. El aspecto del código es el mismo, independientemente del tipo de E/S (USB, GPIB, VXI, RS-232, etc.). Por ejemplo, si estuviéramos utilizando Visual Basic 6.0 y quisiéramos abrir una sesión con un dispositivo SIDL utilizando la dirección de dispositivo especificada por la cadena de dirección de nuestro osciloscopio, necesitaríamos el siguiente código:

```
scope_id=iopen(dirección_del_osciloscopio)
txtStatus.Text = "iopen - no hay errores"
```

SIDL no es el único software de E/S que existe actualmente. AGILENT VISA, NI-VISA y NI-488, así como VISA COM (de Agilent), desempeñan funciones similares. Mientras que el software SIDL se creó para comunicarse únicamente con instrumentos de Agilent, VISA fue creado para funcionar con equipos de todos los fabricantes.

VISA: software de E/S para conseguir la independencia con respecto del hardware y de la interfaz

A finales de los 80, el sector se decantó hacia la creación de instrumentos alojados en armazones de tarjetas estandarizadas. HP (Agilent) lideró un estándar de software y hardware denominado VXI (VME Extensions for Instrumentation). Basado en el estándar VME, VXI introdujo varias modificaciones especiales en cuanto al software, el apantallamiento, el disparo, las fuentes de alimentación y el funcionamiento en modo analógico. Cientos de fabricantes de instrumentos que producían una amplia variedad de dispositivos y tarjetas enchufables adoptaron VXI como estándar. La intercambiabilidad a nivel de tarjeta que aportaba VXI hizo surgir la necesidad de un software de E/S común, similar al lenguaje SIDL de HP, pero implementado

como un estándar aplicable en todo el sector. Así nació la sintaxis VISA, derivada en gran medida de la librería SIDL.

Bajo el liderazgo inicial de National Instruments, se creó la arquitectura de software para instrumentos virtuales (VISA, Virtual Instrument Software Architecture) por la VXIplug&play Foundation, con el objetivo de estandarizar el software de E/S entre las distintas interfaces físicas y los diferentes fabricantes. En la mayoría de los casos, los sistemas de prueba no son únicamente VXI, sino más bien híbridos de arquitecturas VXI e instrumentos apilados en bastidor (Rack & Stack), por lo que no bastaba con crear un software de E/S exclusivamente para VXI. Por este motivo, las especificaciones VXIplug&play se ampliaron para incluir instrumentos autónomos "tradicionales", además de los instrumentos VXI. En la actualidad, los dos principales fabricantes de instrumentos VISA son Agilent Technologies y National Instruments.

VISA (Virtual Instrument Software Architecture, o Arquitectura de software de instrumentos virtuales)

VISA es un estándar de software estándar de E/S creado por la VXIplug&play Foundation. Este software de E/S de VISA usa una terminología y sintaxis comunes para conectarse con y controlar los instrumentos. Para la mayor parte, se trata de un superjuego de SIDL. El estándar VISA sirve para proporcionar un juego común de llamadas a funciones similares en todas las interfaces físicas. En la práctica, las bibliotecas VISA tienden a ser específicas para la interfaz de cada proveedor.

Una librería VISA proporciona un control absoluto sobre los instrumentos a través de las interfaces físicas GPIB, RS-232, USB, LAN y VXI.

El ejemplo en C# que se muestra a continuación abre el Gestor de Recursos (Resource Manager) VISA predeterminado, comprueba que no hay ningún error, y debe invocarse antes de llamar a cualquier otra función VISA:

```
viError = visa32.viOpenDefaultRM(out
resourceManager);
```

Para hacer lo mismo en VB .NET, la sintaxis sería casi idéntica:

```
viError = visa32.viOpenDefaultRM (resource
Manager)
```

La librería VISA ofrece la mayor parte de las posibilidades de SICL, pero siguiendo un método acorde con estándares del sector. Un programa escrito para funcionar con una librería VISA de Agilent funcionará con las implementaciones de VISA de otros fabricantes, por ejemplo NI VISA. Puesto que VISA es casi un superconjunto de la librería SICL, SICL está siendo abandonada gradualmente, en favor del nuevo estándar de librerías VISA adoptado en el sector.

Utilización simultánea de instrumentos de Agilent y de National Instruments

Es muy frecuente que deban coexistir elementos de hardware y software de distintos fabricantes para que un sistema de prueba funcione correctamente. Para que varios productos de distintos fabricantes puedan funcionar juntos puede ser necesario realizar ciertos procedimientos especiales de configuración, y la depuración y resolución de problemas en este tipo de sistemas integrados por equipos de diferentes fabricantes puede resultar más complicada.

¿El orden de instalación de su software de E/S le bloquea otras opciones?

National Instruments

- Si ya tiene instalado Agilent VISA en su sistema e instala posteriormente NI-VISA 3.0, necesitará eliminar completamente Agilent VISA para que funcione.
- Con NI-VISA 3.1, si ya está instalado Agilent VISA, NI cambiará el nombre de Agilent VISA por `visa32.AgilentTechnologies.dll`

Agilent Technologies

- IO Libraries Suite 14.0 detecta si está instalado NI-VISA y selecciona automáticamente Agilent VISA como secundario (esta opción puede cambiarse)

Orden de instalación recomendado

1. Instalar primero NI VISA
2. Después instalar Agilent VISA a través de IO Libraries Suite

El estándar VISA requiere que esté instalada la librería de enlace dinámico `visa32.dll`, donde está implementada la interfaz VISA, en varios lugares prescritos; por tanto, en cada momento sólo puede haber una implementación instalada (la implementación primaria) que cumpla íntegramente el estándar.

He aquí algunos aspectos que hay que tener en cuenta:

- Agilent instala Agilent Connection Expert e Interactive IO, que sólo están disponibles en la familia de librerías de E/S Agilent IO Libraries Suite, al instalar Agilent VISA. Análogamente, NI instala NI Measurement & Automation Explorer (MAX) y VISA Interactive Control (VISAIC) con NI-VISA.

- No es necesario instalar NI-VISA para poder utilizar tarjetas y dispositivos GPIB de National Instruments. Puede instalarse NI-488.2 como controlador para esos dispositivos, y evitar las complicaciones que conlleva utilizar los equipos en modo adosado ("side-by-side"). La librería Agilent IO Libraries Suite es capaz de reconocer estos dispositivos.

- Si su aplicación exige que coexistan Agilent VISA y NI-VISA, puede instalar Agilent VISA en modo "side-by-side" para evitar problemas de compatibilidad de software. La mayoría de los dispositivos y aplicaciones funcionan correctamente en esta configuración.

- Para que el software de NI reconozca el hardware Agilent, es necesario habilitar NI-VISA Passport for Tulip. Se trata de un módulo de software que viene incluido en NI-VISA para permitir el uso de hardware Agilent, y que no está habilitado de forma predeterminada.

- Instale primero NI VISA y después Agilent IO Libraries Suite (Ver barra lateral).

La tecnología de PC permite independizar el software de E/S del lenguaje de programación utilizado

Mientras la industria de los equipos de medida y prueba iba desarrollando su E/S, la industria del PC avanzaba a pasos agigantados hacia la independencia del lenguaje de programación y del sistema de E/S. En 1994, Microsoft declaró: "El modelo COM (Component Object Model) es una arquitectura de software que permite que distintos componentes creados por diferentes proveedores de software puedan combinarse en una gran variedad de aplicaciones. COM define un estándar de interoperabilidad entre componentes, no depende de ningún lenguaje de programación concreto, está disponible sobre varias plataformas distintas y es ampliable."²

El concepto de componentes de software es análogo al de los componentes de hardware; dos componentes que presenten la misma interfaz y desempeñen las mismas funciones se consideran intercambiables. Los componentes de software son la extensión natural del concepto de DLL. Los programas que enlazan con una DLL no pueden utilizar varias implementaciones distintas al mismo tiempo, pero en un mismo ordenador pueden residir varios servidores COM que dispongan de los mismos métodos y puedan utilizarlos diferentes aplicaciones. Además, aunque la funcionalidad básica de una DLL puede ser la misma de un lenguaje a otro, el mecanismo de interfaz de las DLL no está bien definido y puede presentar comportamientos diferentes según el lenguaje de que se trate, por lo que un cambio de entorno de desarrollo obliga a repetir las pruebas y la integración.

Microsoft desarrolló el estándar COM para permitir a los fabricantes crear nuevos componentes de software que pudieran utilizarse con un

programa de aplicación existente sin necesidad de volver a compilar la aplicación. Esta capacidad hace posible intercambiar instrumentos de medida y prueba y su software basado en COM.

VISA COM

La librería VISA COM es una interfaz COM para el software de E/S que sirve de complemento a la especificación VISA. La E/S que ofrece VISA COM proporciona los mismos servicios de VISA a través de una API basada en COM. VISA COM incluye varios servicios de alto nivel que no están disponibles en VISA pero, en cuanto a funcionalidades de comunicación de E/S de bajo nivel, VISA COM es un subconjunto de VISA. Agilent VISA COM es el sistema utilizado por sus controladores IVI-COMt, y necesita que esté instalado también Agilent VISA.

En febrero de 2001, Microsoft introdujo el entorno .NET, su última generación de tecnología de componentes. El marco .NET emplea una interfaz de programación orientada a objetos que simplifica el desarrollo en un entorno Windows.

El entorno .NET Framework tiene dos componentes principales: el módulo de ejecución común ("runtime") del lenguaje y la librería de clases .NET. Este entorno se ha aplicado a Visual Studio .NET, MS Office, además de otras aplicaciones, sistemas operativos y servicios web.

Los ejemplos de comandos VISA COM que se muestran a continuación ilustran la independencia de los comandos con respecto al lenguaje.

Por ejemplo, si estuviéramos utilizando VB .NET y quisiéramos conocer la versión de hardware de un instrumento (en este caso, un multímetro digital), el código sería el siguiente:

```
ioDmm.WriteString(":Sys:Vers?")
result = ioDmm.ReadString
```

Para hacer lo mismo con el más reciente lenguaje de Microsoft, C#, la sintaxis sería la siguiente:

```
ioDmm.WriteString("SYST:VERS?", true);
result = ioDmm.ReadString();
```

Dónde encontrar software de E/S

Puede encontrar el software de E/S VISA de NI en la website de National Instruments, en la dirección <http://www.ni.com/visa>. En diciembre de 2004, este producto podía adquirirse por 395 USD, o venía incluido con la compra de otros productos de NI.

Puede encontrar el software de E/S VISA o VISA COM de Agilent en www.agilent.com/find/iosuite, dentro del área de descarga de la familia de librerías de E/S Agilent IO Libraries Suite. Puede descargarse gratuitamente una copia por cada tarjeta de E/S, instrumento o software de Agilent que haya adquirido.

Controladores de instrumentos

Para poder comunicarse con un instrumento, necesitará aprenderse o consultar los comandos SCPI correspondientes al instrumento concreto que esté programando. A medida que aumenta la complejidad de una aplicación, se vuelve más difícil y engorroso utilizar los comandos nativos del instrumento. Para programar una ruta de comunicación directa suele hacer falta una cierta familiaridad con las secuencias de comandos adecuadas y con las interrelaciones entre los diferentes comandos. También es preciso saber cómo cargar y configurar diversas librerías de E/S estáticas y dinámicas e interpretar las respuestas de los instrumentos, que pueden venir en formato de datos binarios o como gráficos de pantalla. Por eso se crearon los controladores de instrumentos: para simplificar la comunicación.

En el mundo de los sistemas de medida y prueba, los controladores facilitan la comunicación con un instrumento de varias maneras. Un controlador puede generar directamente código de prueba, o bien orientar al usuario por los distintos pasos del

procedimiento. Su interfaz de usuario puede presentar muchas formas distintas. Podría ser una lista de funciones del instrumento que se muestre por medio de la función Intellisense de Microsoft al programar en Visual Basic o en C++, o algo tan elaborado como un "panel virtual" que muestre una representación simulada del panel frontal del instrumento en la pantalla del ordenador, para ayudarle a configurar el instrumento.

Controlador de instrumento

Un controlador de instrumento es un conjunto de funciones que reside en un ordenador y se utiliza para controlar un dispositivo periférico. Se trata de un componente de software de alto nivel específico de un determinado instrumento o clase de instrumentos, cuya misión es facilitar la programación y acortar los tiempos de desarrollo.

Incluso aunque no haya programado nunca ningún instrumento de un sistema de pruebas, probablemente sí haya utilizado un controlador. Cámaras digitales, discos duros externos e impresoras.. todos ellos necesitan un controlador para comunicarse con el PC. Si ha actualizado alguna vez un PC, quizás se haya encontrado con que el controlador de impresora antiguo ya no funciona con el nuevo sistema operativo, y haya necesitado descargarse uno nuevo. Con los equipos de medida y prueba ocurren cosas similares.

Los controladores incluyen entre el 40 y el 60% de las funciones comunes de los instrumentos

Un voltímetro sencillo puede tener sólo 25 comandos, mientras que uno más complejo puede llegar a tener centenares. Debido al elevado coste de desarrollo que supone, es muy raro que un controlador incluya todos y cada uno de los comandos de un instrumento. Según los

diversos análisis que ha realizado Agilent, un controlador de instrumento típico suele abarcar entre el 40 y el 60% de toda la funcionalidad de un instrumento. En realidad, para muchos usuarios eso es más que suficiente. Por ejemplo, en una encuesta realizada por Agilent entre clientes que habían utilizado uno de nuestros instrumentos más complejos, con más de 300 comandos, descubrimos que muy pocos habían usado más del 5% de los comandos disponibles. Si necesita alguna funcionalidad del instrumento que no esté disponible en el controlador, puede acceder a ella mediante la entrada/salida directa.

Resumen

Para poder comunicar el PC con los instrumentos, debe tener instalado software de E/S. La versión más

reciente de la especificación VISA es fruto de varias generaciones de software de E/S. El software de E/S más utilizado es el VISA/VISA COM de Agilent y el NI-VISA/VISA COM de NI. Para disfrutar de la máxima variedad de opciones si necesita trabajar software de E/S de varios fabricantes distintos, lo mejor es que instale primero NI VISA y después Agilent VISA, ya que Agilent IO Libraries Suite detectará la presencia de todo el software de E/S y se instalará de modo que sea posible utilizar complementariamente cualquiera de los dos softwares de E/S, o ambos. No es necesario que el fabricante del software de E/S sea el mismo que el del hardware, ya se trate de instrumentos o de tarjetas de adquisición de datos.

Utilizar los comandos nativos del instrumento, con independencia de la interfaz con el instrumento que se use, proporciona un acceso total a

toda la funcionalidad del instrumento, y permite ejecutar las instrucciones a la máxima velocidad posible.

Los controladores de instrumentos permiten acceder fácilmente a un repertorio de entre el 40 y el 60 % de toda la funcionalidad del instrumento, y es frecuente utilizar tanto controladores como comandos nativos para controlar un instrumento. o

Notas

(1) Toda la información ha sido verificada, hasta donde el autor ha sido capaz, para garantizar su veracidad y precisión a 15 de diciembre de 2004. Sin embargo, la tecnología está cambiando constantemente, por lo que antes de tomar cualquier decisión, es conveniente investigar todos los detalles.

(2) Dr. Dobb's Journal, Microsoft Corp. Diciembre de 1994 .