

**UNIVERSIDAD POLITÉCNICA DE VALENCIA**

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE  
TELECOMUNICACIÓN**

**PRACTICA:**

**CONTROL  
DE  
INSTRUMENTOS  
CON  
ARQUITECTURA VISA  
·r3**

**Autor:**  
**José M<sup>a</sup> Grima Palop**

**Departamento de Ingeniería Electrónica**



## ÍNDICE

<b>1. INTRODUCCIÓN .....</b>	<b>5</b>
<b>2. MATERIAL NECESARIO .....</b>	<b>5</b>
<b>3. VISA.....</b>	<b>5</b>
3.1. TEORÍA .....	5
3.2. APLICACIÓN.....	8
<b>4. DRIVER DE INSTRUMENTO .....</b>	<b>8</b>
4.1. TEORÍA .....	8
4.2. APLICACIÓN.....	10
<b>5. HOJA DE RESULTADOS .....</b>	<b>11</b>



## 1. INTRODUCCIÓN

En la presente práctica se controlará el multímetro digital AGILENT 34401A desde LabVIEW con funciones VISA donde se realice una búsqueda automatizada del multímetro y por último se trabajará con el driver del instrumento para realizar una serie de medidas automatizadas. Previamente se aprenderá a gestionar la cadena de búsqueda de recursos.

## 2. MATERIAL NECESARIO

Para la realización de la práctica, será necesario disponer;

- ✓ Multímetro digital Agilent 34401A.
- ✓ PC con software LabVIEW.
- ✓ Década de resistencias.
- ✓ 2 cables banana-banana.

## 3. VISA

### 3.1. TEORÍA

La Alianza de Sistemas *VXIplug&play* era una organización cuyos miembros compartían un compromiso con los usuarios finales de sistemas abiertos VXI. La Alianza logró una mejora en la facilidad de uso de los sistemas de instrumentación, gracias a la incorporación de estándares y experiencia tanto en el hardware como el software, que van más allá de las especificaciones del bus VXI. El empleo oficial de estándares *de facto* ofrece a los usuarios una interoperabilidad *plug&play* tanto en hardware como en software.

El mayor éxito de *VXIplug&play* fue ofrecer una mayor facilidad de uso sin modificar la arquitectura abierta y multi-proveedor de VXI. Cualquier hardware VXI, compatible con las especificaciones VXIbus es susceptible de ser incluido en un entorno de trabajo *VXIplug&play*. Por tanto los fabricantes no necesitan ser miembros de la organización para que sus productos sean compatibles.

La Alianza ofrecía una interoperabilidad *plug&play* debido a que identifica y promueve entornos software de trabajo estándares. Esos entornos de trabajo son sistemas operativos, lenguajes de programación, drivers de E/S, drivers de instrumentos y herramientas software de aplicación de alto nivel. Concretamente la Alianza consideraba obligatorio suministrar junto con el instrumento correspondiente un driver de instrumento, tanto para programar como para usar interactivamente. Adicionalmente consideraba los elementos estándares de software como DLL, ActiveX/COM/DCOM, DDE, etc., como posibles vías de ampliación y por tanto pueden coexistir en el mismo sistema.

Para alcanzar la funcionalidad *plug&play* tan deseada, es necesaria una arquitectura unificada de driver de E/S. *VXIplug&play* definió VISA (*Virtual Instrument Software Architecture*) como la única arquitectura de driver de E/S para implementar todos los aspectos que ofrecía *VXIplug&play* en el presente y en el futuro.

La Alianza de sistemas *VXIplug&play* reconoció diez puntos que deben guiar su actividad, que son:

- i. **Maximizar la facilidad de uso y las prestaciones.** La facilidad de uso ha sido desde el origen el interés principal de la Alianza. La cuestión consiste en que para facilitar

todavía más el uso, hay que facilitar la integración. Algo que tarde o temprano revierte en un reducido coste de desarrollo y mantenimiento.

- ii. **Mantenimiento de la compatibilidad con el sistema inicial.** *VXIplug&play* mantiene la compatibilidad regresiva con instrumentos ya existentes o con sistemas previamente integrados. De la misma forma asegura que todo usuario que construya su sistema conforme con el entorno de trabajo propuesto por *VXIplug&play*, consolidará su inversión de cara al futuro.
- iii. **Mantener la arquitectura abierta multi-proveedor.** Cualquier entorno de trabajo implementado de acuerdo con *VXIplug&play* puede emplear instrumentos VXI y GPIB de cualquier fabricante, sin tener en cuenta si el instrumento es compatible con los entornos *VXIplug&play*.
- iv. **Maximizar la capacidad multiplataforma.** Se pretende que el usuario final decida libremente la plataforma sobre la que desea desarrollar su aplicación.
- v. **Maximizar la modularidad y la ampliabilidad en las arquitecturas de sistemas.** El entorno de trabajo *VXIplug&play* es ampliable y modular de tal forma que los usuarios pueden mantener su sistema durante todo su ciclo de vida.
- vi. **Maximizar el software reutilizable.** *VXIplug&play* indica las pautas a seguir para maximizar la reutilización del software por parte del fabricante como del usuario final.
- vii. **Estandarizar el uso de elementos software del entorno.** *VXIplug&play* identifica y adopta software estándar como drivers de E/S, sistemas operativos, lenguajes de programación, drivers de instrumentos y paneles software.
- viii. **Tratar al driver del instrumento como una parte del mismo.** *VXIplug&play* emplea drivers tanto para trabajar con lenguajes de programación estándar, como para realizar módulos de aplicación software. El driver de instrumento es uno de los componentes que aportan un mayor valor añadido al sistema, aportando una mayor facilidad de uso y de desarrollo. Los drivers de instrumento *VXIplug&play* están completamente verificados, documentados, abiertos y cuando es posible, se distribuyen en código fuente. La Alianza ha desarrollado un estándar común para el desarrollo, arquitectura, distribución y soporte de drivers de instrumentos. El driver del instrumento debería desarrollarlo el propio fabricante. No obstante, en el caso de que no sea él el que lo desarrolle no impide su compatibilidad. Las herramientas y técnicas para desarrollar drivers de instrumento se encuentran disponibles para cualquier usuario.
- ix. **Adoptar estándares establecidos.** Siempre que sea posible, *VXIplug&play* empleará estándares consolidados. No obstante, la Alianza podrá definir estándares adicionales en tecnología, los productos y los servicios siempre y cuando supongan una mayor facilidad de uso, un tiempo de desarrollo y coste de mantenimiento menor o una mayor compatibilidad.
- x. **Maximizar el soporte corporativo a los usuarios finales.** *VXIplug&play* incrementa el nivel de cooperación entre vendedores y los usuarios finales en lo que se refiere al soporte del producto. Para conseguirlo se realizan encuentros entre personal técnico de soporte, programas de actualización del personal, etc.

Actualmente *VXIplug&play* junto con VISA forman parte de IVI Foundation.

Los objetos a controlar se les designa como recursos y la forma de identificarlos es mediante una cadena de direccionamiento cuya gramática está estructurada y se muestra en la Tabla 3-1.

Tabla 3-1

Interface	Sintaxis
VXI	VXI[ <i>board</i> ]::VXI logical address[:INSTR]
VXI	VXI[ <i>board</i> ]::MEMACC
VXI	VXI[ <i>board</i> ]::VXI logical address[:BACKPLANE]
VXI	VXI[ <i>board</i> ]::SERVANT
GPIB-VXI	GPIB-VXI[ <i>board</i> ]::VXI logical address[:INSTR]
GPIB-VXI	GPIB-VXI[ <i>board</i> ]::MEMACC
GPIB-VXI	GPIB-VXI[ <i>board</i> ]::VXI logical address[:BACKPLANE]
GPIB	GPIB[ <i>board</i> ]::primary address[:secondary address[:INSTR]]
GPIB	GPIB[ <i>board</i> ]::INTFC
GPIB	GPIB[ <i>board</i> ]::SERVANT
ASRL	ASRL[ <i>board</i> ]::INSTR
TCPIP	TCPIP[ <i>board</i> ]::LAN device name[:SERVANT]
TCPIP	TCPIP[ <i>board</i> ]::host address[:LAN device name[:INSTR]]
TCPIP	TCPIP[ <i>board</i> ]::host address[:HiSLIP device name[,HiSLIP port]][:INSTR]
TCPIP	TCPIP[ <i>board</i> ]::host address::port::SOCKET
USB	USB[ <i>board</i> ]::manufacturer ID::model code::serial number[:USB interface number[:INSTR]]
PXI	PXI[ <i>bus</i> ]::device[:function[:INSTR]]
PXI	PXI[ <i>interface</i> ]::bus-device[:function[:INSTR]]
PXI	PXI[ <i>interface</i> ]::CHASSISchassis::SLOTslot[:FUNCfunction[:INSTR]]
PXI	PXI[ <i>interface</i> ]::MEMACC
PXI	PXI[ <i>interface</i> ]::chassis number::BACKPLANE

Los valores por defecto son los indicados en la tabla siguiente:

Optional String Segment	Default Value
board	0
GPIB secondary address	none
LAN device name	inst0
HiSLIP device name	hislip0
HiSLIP port	4880
USB interface number	lowest numbered relevant interface
PCI function number	0

VISA incluye una función especialmente dedicada a la búsqueda de recursos que se puede parametrizar empleando la sintaxis que se recoge en la siguiente tabla:

Special Characters and Operators	Meaning
?	Matches any one character.
\	Makes the character that follows it an ordinary character instead of special character. For example, when a question mark follows a backslash (\?), it matches the ? character instead of any one character.
[list]	Matches any one character from the enclosed list. You can use a hyphen to match a range of characters.
[^list]	Matches any character not in the enclosed list. You can use a hyphen to match a range of characters.
*	Matches 0 or more occurrences of the preceding character or expression.
+	Matches 1 or more occurrences of the preceding character or expression.
Exp exp	Matches either the preceding or following expression. The or operator   matches the entire expression that precedes or follows it and not just the character that precedes or follows it. For example, VXI GPIB means (VXI) (GPIB), not VX(I G)PIB.
(exp)	Grouping characters or expressions.

## 3.2. APLICACIÓN

De acuerdo con la información obtenida en las transparencias y en el apartado anterior, se pide;

1. Ejecutar la función “Find Resource” para que detecte todo recurso disponible y copiar en la tabla literalmente los recursos encontrados y describirlos.
2. Escribir en la hoja de resultados la cadena de búsqueda más corta para que, con la función “Find Resource”, obtenga los siguientes recursos VISA: i) todos los instrumentos disponibles. ii) todos los instrumentos asociados a todos los puertos serie del PC. iii) todos los instrumentos conectados a cualquier bus GPIB desde la dirección 0 a la 9 exclusivamente.
3. Acceder a la página web de bienvenida del instrumento LXI conectado por ethernet e insertarla en el cuadro de imagen. Si el instrumento estuviese conectado por USB, indicar cuál sería el nombre del recurso VISA a emplear.
4. Realizar un VI con funciones VISA donde el programa busque de entre todos los instrumentos conectados al bus GPIB, cuál es el multímetro digital HP34401A, presente su cadena de direccionamiento de forma automática en un indicador y se quede midiendo y visualizando de forma continua resistencia en el rango de 10 k $\Omega$  con 5 ½ dígitos hasta que el usuario pulse un botón STOP. (Cuando funcione guardarlo en una librería con nombre “P4-Apellido, nombre.llb”).

## 4. DRIVER DE INSTRUMENTO

### 4.1. TEORÍA

La forma más sencilla de despreocuparse de ciertos detalles de la comunicación con los instrumentos, sin que afecte al funcionamiento del programa, es emplear un driver de instrumento que optimice sus recursos y sea transparente para el usuario.

Un driver de instrumento consiste en una librería de funciones software destinadas a controlar un instrumento específico. *VXIplug&play* indicaba de forma explícita cómo se debe diseñar, desarrollar y distribuir drivers de instrumento multi-proveedores. La Alianza



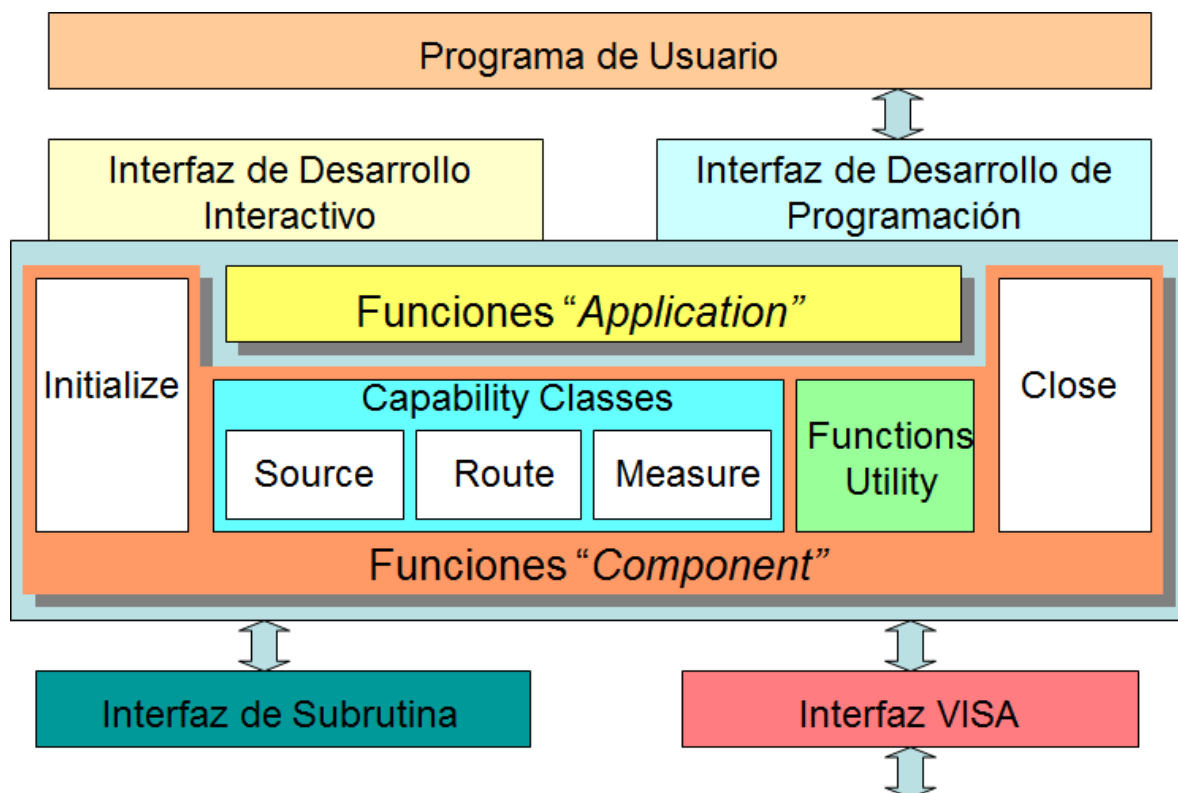
describía un único modelo de arquitectura de driver, así como las líneas generales para que se puedan emplear en una gran variedad de entornos de trabajo. Las ventajas de una única arquitectura de driver son:

- Unifica el proceso de desarrollo del driver.
- Ofrece consistencia entre drivers de diversos fabricantes.
- Mejora la calidad de los mismos.
- Evita la duplicidad de recursos.
- Facilita el uso a los usuarios finales al ofrecer una metodología única.
- Un mismo driver de instrumento puede ser empleado por diferentes ADE.
- Posibilidad de combinar drivers de diversos fabricantes.

Los requisitos que establecía *VXIplug&play* sobre los drivers son:

- a) El cuerpo funcional del driver debe suministrarse en código fuente. Permite al usuario entender, modificar y mejorarlo para adaptarlo a sus necesidades.
- b) Debe ser modular y jerárquico.
- c) Debe ser consistente en el diseño y la implementación. Representa una ventaja ya que todos los desarrolladores de drivers deben trabajar sobre la misma arquitectura. Una vez hecho un driver, el resto es lo mismo.
- d) Deben tener un testigo de errores. Todo driver debe devolver el estado de error, que será del tipo de dato *ViStatus*. Si se produce un error que impide su ejecución devolverá un número identificativo de la causa. Si no hay error, devolverá el valor *VI\_SUCCE*S. Los errores tienen preferencia sobre los mensajes de precaución. Los mensajes de precaución no son causa para detener la ejecución.
- e) Ofrecerán varios niveles de ayuda.
- f) Cualquier función que suministre el driver deberá documentarse con detalle, indicando el entorno de trabajo, parámetros de entrada y de salida.
- g) Indicará la versión del entorno de trabajo con la cual es compatible.

*VXIplug&play* definía dos modelos de arquitecturas de driver. El primero se le conoce como el *modelo de interfaz externo*. En el cual se presentan las vías de comunicación con el resto del software del sistema. El segundo es el *modelo de diseño interno* y describe con detalle la organización interna de un driver. A continuación se detalla esquemáticamente el conjunto de los dos modelos.



## 4.2. APLICACIÓN

5. Realizar un VI con el driver del instrumento que permita seleccionar cualquier función de medida posible por multímetro y realizar medidas de forma continua hasta que el usuario pulse un botón STOP. El VI deberá visualizar cinco valores y presentar el valor medio junto con su desviación típica. (Cuando funcione guardar el archivo en la librería citada y notificarlo al profesor de la práctica).

## 5. HOJA DE RESULTADOS

ALUMNO:

FECHA:

GRUPO:

PUESTO:

1- Copiar todos los recursos encontrados y describirlos con detalle.

RECURSO ENCONTRADO	DESCRIPCION DEL RECURSO

2- Cadena de búsqueda más corta para todos los instrumentos disponibles.

--

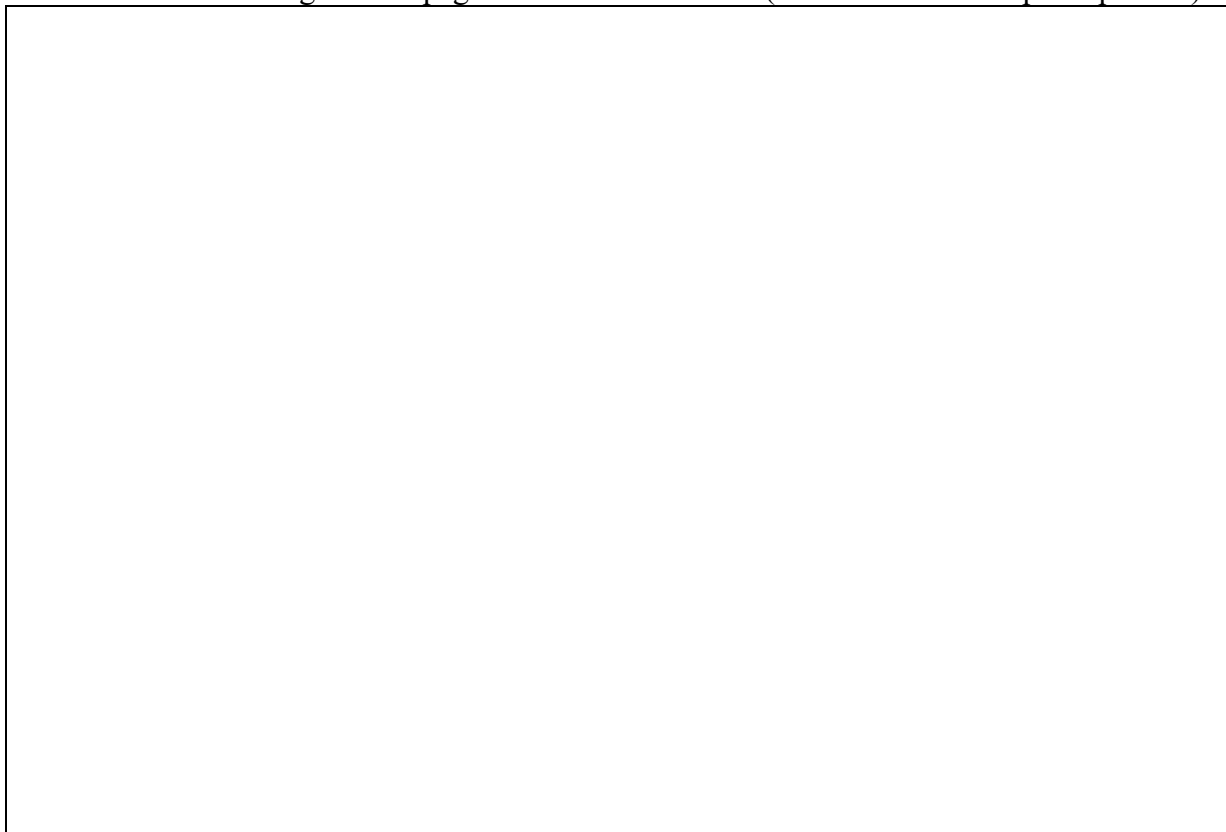
Cadena de búsqueda más corta para todos los instrumentos asociados a los puertos serie del PC.

--

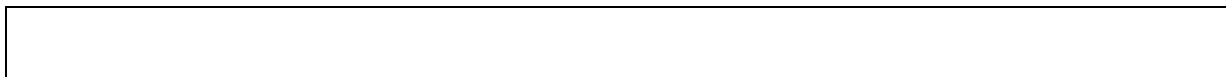
Cadena de búsqueda más corta para todos los instrumentos con direcciones 0 a 9 conectados a cualquier bus GPIB.

--

3- Insertar la imagen de la página web de bienvenida (hacerla lo más compacta posible).



Insertar el nombre del recurso VISA del instrumento LXI en caso de estar conectado por USB.



4- VI con funciones VISA. Insertar el panel frontal con datos reales y el diagrama.



5- VI con driver de instrumento. Insertar el panel frontal con datos reales y el diagrama.


#### PUNTUACION

Descripción de los recursos encontrados.		
Cadena de búsqueda más corta para todos los instrumentos disponibles		
Cadena de búsqueda más corta para todos los instrumentos asociados a los puertos serie del PC.		
Cadena de búsqueda más corta para todos los instrumentos con direcciones 0 a 9 conectados a cualquier bus GPIB		
Página web de bienvenida.		
Nombre del recurso VISA del instrumento LXI en caso de estar conectado por USB.		
VI con funciones VISA. Funcionamiento.		
VI con funciones VISA. Código y panel frontal.		
VI con driver. Funcionamiento.		
VI con driver. Código y panel frontal.		

**BIBLIOGRAFÍA.**

VXIplug&play Systems Alliance. “VPP-4.3: The VISA Library”. 11 octubre 2012.

Transparencias de clase “VISA\_con\_LabVIEW”, por J. M. Grima.