

1.- LA INSTRUMENTACIÓN VIRTUAL

La rápida adopción de los PCs en los últimos 20 años generó una revolución en la instrumentación de ensayos, mediciones y automatización. Un importante desarrollo resultante de la ubicuidad de los PCs es el concepto de *instrumentación virtual*, el cual ofrece variados beneficios a ingenieros y científicos que requieran mayor productividad, precisión y rendimiento.

Un instrumento virtual se fundamenta en un PC, computadora del tipo industrial, o una estación de trabajo, equipada con poderosos programas (software) y hardware, tales como placas para insertar, y manejadores (drivers) que cumplen, en conjunto, las funciones de instrumentos tradicionales. Los instrumentos virtuales representan una evolución fundamental respecto de los sistemas de instrumentación basados en el hardware a sistemas centrados en el software que aprovechan la potencia de cálculo, productividad, exhibición y capacidad de conexión de los populares PCs. Aunque los PCs y la tecnología de circuitos integrados han experimentado avances significativos en las últimas dos décadas, es el software el que realmente provee la ventaja para construir sobre esta potente base de hardware para crear los instrumentos virtuales, proveyendo mejores maneras de innovar y de reducir los costos significativamente. Con los instrumentos virtuales, los ingenieros y científicos construyen sistemas de medición y automatización que se ajustan exactamente a sus necesidades (definidos por el usuario) en lugar de estar limitados por los instrumentos tradicionales de funciones fijas (definidos por el fabricante).

En esta introducción se describen poderosas herramientas de programación, hardware de adquisición de datos flexible y la computadora personal, los cuales son los componentes esenciales de la instrumentación virtual. La sinergia entre ellos ofrece ventajas que no pueden ser igualadas por la instrumentación tradicional.

1.1 Instrumentos virtuales versus Instrumentos tradicionales

Los instrumentos autónomos tradicionales, tales como osciloscopios y generadores de ondas, son muy poderosos, a veces caros, y diseñados para llevar a cabo una o más tareas específicas definidos por el fabricante. Sin embargo, el usuario por lo general no puede extender o personalizar esas tareas. Por ejemplo, los botones del instrumento, sus circuitos electrónicos y las funciones disponibles para el usuario son todas específicas a la naturaleza del instrumento. Además, debe desarrollarse una tecnología especial y costosos componentes para construirlos, lo cual los hace, en general, lentos para adaptarlos a necesidades específicas.

Debido a que están basados en un PC, los instrumentos virtuales aprovechan inherentemente los beneficios de la última tecnología de las computadoras personales corrientes. Estos avances en tecnología y rendimiento, que están cerrando rápidamente la brecha entre los instrumentos autónomos y los PCs, incluyen poderosos procesadores y sistemas operativos. Además de incorporar características poderosas, esas plataformas también ofrecen un acceso sencillo a herramientas también poderosas tales como Internet y conexión inalámbrica. Los instrumentos tradicionales también adolecen frecuentemente

de falta de portabilidad, en tanto que los instrumentos virtuales que corren en las computadoras portátiles automáticamente incorporan esta naturaleza portátil.

Los ingenieros y científicos cuyas necesidades, aplicaciones y requerimientos varían muy rápidamente, necesitan flexibilidad para crear sus propias soluciones, así que se puede adaptar un instrumento virtual a necesidades particulares sin necesidad de reemplazar todo el instrumento dado que posee el software de aplicación instalado en la computadora y al amplio rango disponible de hardware para instalar en ella.

1.2 Flexibilidad

A excepción de los componentes especializados y los circuitos hallados en los instrumentos tradicionales, la arquitectura general de los instrumentos autónomos es muy similar a la hallada en un instrumento virtual basado en computadora. Ambos requieren uno o más microprocesadores, puertos de comunicación (por ejemplo: serie y GPIB) y capacidad de mostrar resultados así como también módulos de adquisición de datos. Lo que diferencia uno del otro es su flexibilidad y el hecho que se puede modificar y adaptar el instrumento a necesidades particulares específicas. Un instrumento tradicional podría contener un circuito integrado para llevar a cabo un conjunto particular de instrucciones de procesamiento de datos; en un instrumento virtual estas funciones podrían llevarse a cabo por el programa que corre en el procesador de la computadora. Así que se puede fácilmente extender ese conjunto de funciones y estar sólo limitado por la potencia del software que se utiliza.

1.3 Reducción de costos

Utilizando soluciones basadas en la instrumentación virtual, se pueden reducir los costos de inversión, desarrollo de sistemas y mantenimiento al mismo tiempo que mejora el tiempo de desarrollo y la calidad de los productos.

1.4 Hardware para insertar y de red

Existe una amplia variedad disponible de hardware que se puede o bien insertar en el PC o bien acceder a través de una red. Estos dispositivos ofrecen un amplio rango de capacidades de adquisición de datos a un costo significativamente inferior que el correspondiente a dispositivos dedicados. A medida que la tecnología de circuitos integrados avanza y los componentes comunes se vuelven más baratos y operativos, también lo hacen las placas que ellos utilizan. Junto con estos avances tecnológicos viene un incremento en las velocidades de adquisición de datos, precisión de las mediciones y mejor aislamiento de las señales.

Dependiendo de la aplicación en particular, el hardware elegido para una aplicación específica, podría incluir entradas o salidas analógicas, entradas o salidas digitales, contadores, temporizadores, filtros, muestreo simultáneo y capacidades de generación de señales de test y medida. La amplia gama de

placas y hardware podría incluir cualquiera de esas características o una combinación de ellas.

1.5 El Software en la Instrumentación Virtual

El software es el componente más importante de un instrumento virtual. Con la herramienta de software apropiada los ingenieros y científicos pueden crear eficientemente sus propias aplicaciones, diseñando e integrando las rutinas que requiere un proceso específico. También pueden crear las interfaces de usuario que mejor satisfagan el objetivo de la aplicación y de aquéllos que van a interactuar con ellas. Pueden definir cómo y cuándo la aplicación adquiere datos desde el dispositivo, cómo los procesa, manipula y almacena los datos y cómo se presentan los resultados al usuario.

Disponiendo de un software poderoso, se puede dotar a los instrumentos con capacidades de inteligencia y de toma de decisiones de manera tal que se adapten cuando las señales medidas varíen inadvertidamente o cuando se requiera mayor o menor potencia de procesamiento.

Una importante ventaja que provee el software es la modularidad. Cuando se trata de un gran proyecto, el equipo de trabajo generalmente aborda la tarea dividiéndola en unidades funcionales manejables (metodología top-down). Estas tareas subsidiarias son más manejables y más fáciles de probar dadas las menores dependencias que podrían causar comportamientos inesperados. En este contexto se puede diseñar un instrumento virtual para solucionar cada una de estas tareas subsidiarias y luego reunir las en un sistema completo para resolver la tarea de mayor envergadura. La facilidad con la cual se puede realizar esta división de tarea depende en mayor medida de la arquitectura subyacente en el software.

1.6 Aplicaciones distribuidas

Un instrumento virtual no está limitado a estar confinado en una computadora autónoma. En realidad, con los recientes desarrollos en tecnologías de redes e Internet, es más común utilizar la potencia de conectividad de los instrumentos con el fin de compartir tareas. Ejemplos típicos incluyen las supercomputadoras, monitorización distribuida y dispositivos de control, así como también datos o visualización de resultados desde múltiples emplazamientos.

1.7 LabVIEW: Herramienta para desarrollar Instrumentos Virtuales

LabVIEW es una parte integral de la instrumentación virtual dado que provee un entorno de desarrollo de aplicaciones que es fácil de utilizar y está diseñado específicamente teniendo en mente las necesidades operativas y funcionales asociadas a la ingeniería electrónica. LabVIEW ofrece poderosas características que facilitan la conexión a una gran variedad de hardware y otros entornos software.

1.8 Programación gráfica

Una de las características más poderosas que LabVIEW ofrece es un entorno de programación que es gráfico. Con LabVIEW se pueden diseñar instrumentos virtuales a medida creando interfaces gráficas de usuario en la pantalla de la computadora con la cual es posible:

- Operar con el programa de instrumentación.
- Controlar el hardware seleccionado.
- Analizar datos adquiridos.
- Visualizar los resultados.

Se pueden personalizar paneles frontales con botones, diales y gráficos a fin de emular paneles de control de instrumentos tradicionales, crear paneles de ensayo personalizados o representar visualmente el control y operación de procesos. La similitud existente entre los diagramas de flujo y los programas gráficos acorta la curva de aprendizaje asociada con lenguajes tradicionales basados en texto.

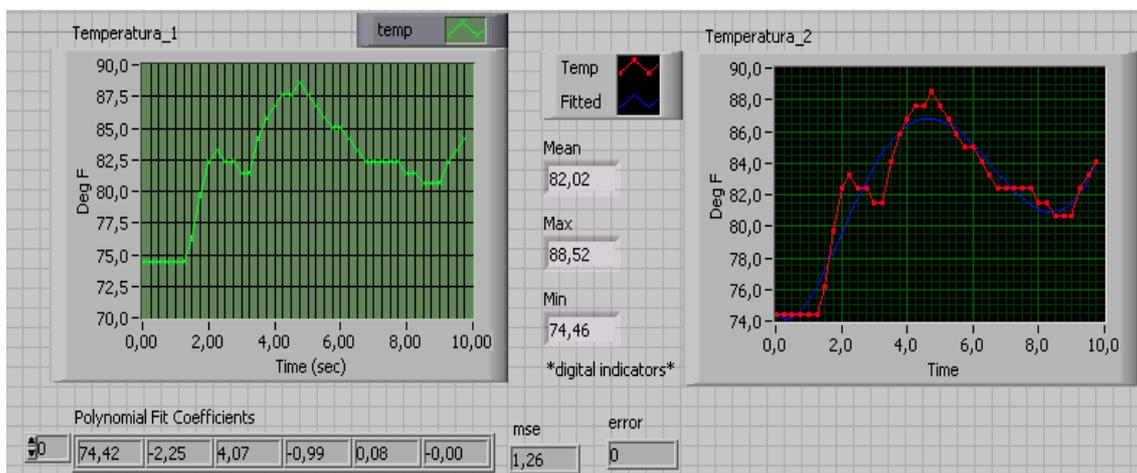


Figura 1. Panel Frontal de un Instrumento Virtual hecho con LabVIEW

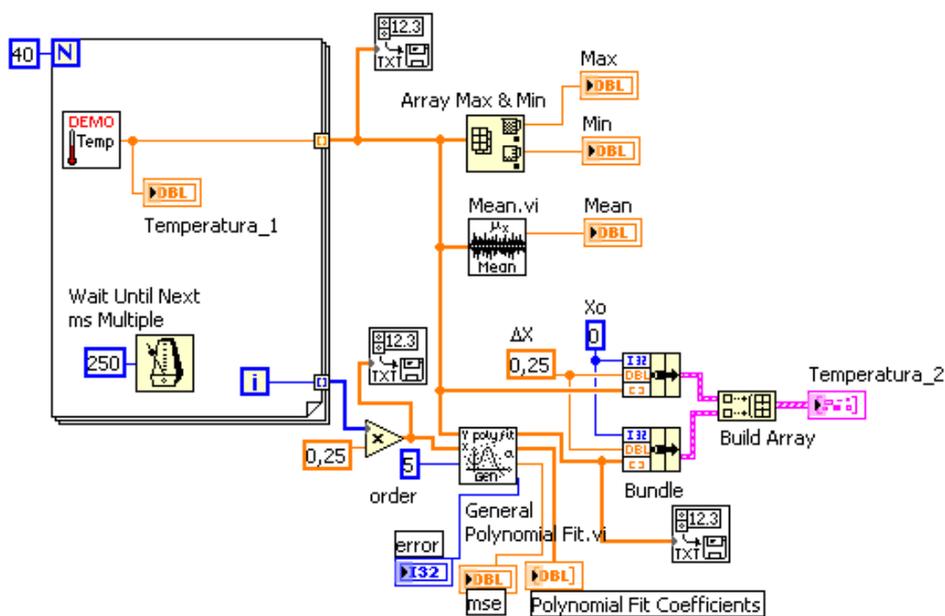


Figura 2. Diagrama de Bloques de un Instrumento Virtual hecho con LabVIEW

Es posible determinar el comportamiento de los instrumentos virtuales conectando íconos entre sí para crear diagramas de bloques, que son notaciones de diseño naturales para ingenieros y científicos. Con un lenguaje gráfico el desarrollo de los sistemas de instrumentación es más rápido que con lenguajes de programación convencionales de tipo texto, mientras que conserva la potencia y flexibilidad necesarias para crear una amplia variedad de aplicaciones.

1.9 Conectividad y control de instrumentos

La productividad del software de instrumentación virtual es tal ya que incluye el conocimiento de la integración del hardware. Diseñado para crear ensayos, mediciones y control de sistemas, el software de la instrumentación virtual incluye una extensa funcionalidad para entradas y salidas prácticamente de cualquier tipo.

LabVIEW posee bibliotecas listas para ser utilizadas con el objeto de integrar instrumentos autónomos, equipos de adquisición de datos, productos para el control de movimientos y de visión, instrumentos GPIB/IEEE 488 y serie RS-232 y PLCs, conexiones inalámbricas ,etc, lo cual permite construir una solución completa de medición y automatización. LabVIEW también tiene incorporadas las más importantes normas de instrumentación, tal como VISA, una norma que permite la operación entre instrumentos GPIB, serie y VXI; PXI y software y hardware basados en la norma PXI Systems Alliance Compact PCI; manejadores de instrumentos virtuales intercambiables IVI y VXI *Plug&Play*, que es un manejador para la norma que rige la instrumentación VXI.

1.10 Plataforma abierta

Aunque LabVIEW provee las herramientas requeridas para la mayoría de las aplicaciones, también constituye un entorno de desarrollo abierto. El cumplimiento de normas por parte del software se basa principalmente en la funcionalidad del entorno elegido para trabajar bien junto a otros sistemas de software y hardware de medición y control, y normas abiertas, que definen la capacidad de interactuar entre múltiples fabricantes, de esta forma se aprovechan los productos ofrecidos por numerosos proveedores. Además, ajustándose a normas abiertas, se reduce el costo total del sistema.

Un gran número de fabricantes de hardware y software desarrollan y mantienen centenares de bibliotecas de LabVIEW y drivers de instrumentos que ayudan a utilizar fácilmente sus productos con LabVIEW. Sin embargo, esa no es la única manera de proveer conectividad a las aplicaciones basadas en LabVIEW.

Este software ofrece maneras simples de incorporar programas en ActiveX, bibliotecas dinámicas (DLLs) y bibliotecas compartidas de otras herramientas. Además, se puede compartir código hecho en LabVIEW como una DLL, construir un programa ejecutable o utilizar ActiveX.

LabVIEW también ofrece un amplio abanico de opciones para comunicaciones tales como, TCP/IP, OPC, SQL y formato XML.

LabVIEW es un producto versátil dado que se puede utilizar un solo PC equipado con LabVIEW para innumerables aplicaciones y propósitos. No sólo es versátil sino también extremadamente efectivo desde el punto de vista del costo. A medida que cambian las necesidades operativas del sistema implementado, se puede fácilmente modificar los sistemas sin necesidad de adquirir nuevo equipamiento y crear bibliotecas enteras de instrumentación a costo menor que el correspondiente a un solo instrumento comercial tradicional.

1.11 Plataformas múltiples

La mayoría de los sistemas computacionales utilizan alguna variante del sistema operativo Microsoft Windows; no obstante, existen otras opciones que ofrecen claras ventajas para ciertos tipos de aplicaciones. El desarrollo de sistemas operativos de **tiempo real** y **embebido** continúa creciendo rápidamente en la mayoría de las industrias a medida que la capacidad de cálculo es incorporada en paquetes más especializados y pequeños. Es importante minimizar las pérdidas resultantes del cambio hacia nuevas plataformas y la elección del software correcto para dicho objetivo es un factor clave.

LabVIEW minimiza esta preocupación ya que corre en Windows 2000, NT, XP, Me, 98, 95 y NT embebido así como también sobre Mac OS, Sun Solares y Linux. Con LabVIEW también se puede compilar código que corra en el sistema operativo de tiempo real VenturCom ETS a través del módulo LabVIEW Real-Time. LabVIEW es independiente de la plataforma seleccionada: los instrumentos virtuales creados para una plataforma pueden ser transportados de manera transparente a cualquier otra plataforma simplemente abriendo el instrumento virtual.

Puesto que las aplicaciones de LabVIEW pueden transportarse entre plataformas, se asegura que el trabajo de hoy será utilizable en el futuro. A medida que emerjan nuevas tecnologías computacionales, es factible migrar fácilmente las aplicaciones a las nuevas plataformas y sistemas operativos. Además, debido a la implementación de instrumentos virtuales que son independientes de la plataforma y transportarlo entre distintas plataformas, se ahorra tiempo de desarrollo y otros inconvenientes relacionados con la portabilidad entre plataformas.

1.12 Desarrollo distribuido

Con LabVIEW se implementan fácilmente aplicaciones distribuidas, aún entre diferentes plataformas. Con herramientas de servidores fáciles de usar, se pueden descargar rutinas que hacen un uso intensivo del procesador a otras máquinas para lograr una ejecución más rápida, o crear aplicaciones de monitorización y control remoto. Una tecnología de servidores poderosa puede simplificar la tarea de desarrollar aplicaciones grandes y que requieran de múltiples computadoras. Además de ello, LabVIEW incluye tecnologías

normalizadas de redes, tales como TCP/IP e incorpora protocolos robustos de publicación y suscripción.

1.13 Capacidades de análisis

El software de la instrumentación virtual requiere de complejas herramientas de análisis y procesamiento de señales ya que la aplicación no se detiene justo cuando el dato es recogido. Usualmente, las aplicaciones de mediciones de alta velocidad en los sistemas de monitorización y control requieren análisis detallados y precisos para lograr obtener datos relativos a la evolución del proceso. Los **sistemas de control embebidos de lazo cerrado** podrían requerir promediación punto a punto para lograr que los algoritmos de control mantengan la estabilidad. A modo de ejemplo, además de las bibliotecas de análisis avanzado ya incluidas en LabVIEW, National Instruments provee software adicional, tal como el LabVIEW Signal Processing Toolset (Paquete de Herramientas para el Procesamiento de Señales LabVIEW) a fin de complementar las ofertas de análisis.

1.14 Capacidades de visualización

LabVIEW incluye un amplio conjunto de herramientas de visualización para presentar datos en la interface de usuario, tanto para gráficos continuos como para la visualización de gráficos 2D y 3D. También se puede reconfigurar de manera instantánea los atributos de la presentación de los datos, tales como: colores, tamaño de fuentes, tipos de gráfico y más, así como también efectuar rotación, enfoque (zoom) y desplazamiento dinámico en estos gráficos con el ratón.

1.15 Flexibilidad y escalabilidad – Ventajas clave

La ingeniería tiene necesidades y requerimientos que cambian rápidamente, por lo tanto, es necesario disponer de soluciones extensas que puedan ser mantenidas y utilizadas por un largo período. Creando instrumentos virtuales basados en un software de desarrollo amplio, como es LabVIEW, se diseña en un entorno de trabajo abierto que se integre de modo ininterrumpido con el software y el hardware. Esto asegura que las aplicaciones no sólo funcionarán sino que también se podrá fácilmente integrar nuevas tecnologías en el futuro a medida que se hallen disponibles, o extender sus soluciones más allá de su alcance original a medida que se identifiquen nuevos requerimientos. Más aún, cada aplicación posee sus propios y únicos requerimientos que precisan una amplia gama de soluciones.

1.16 La Instrumentación Virtual dentro del proceso de ingeniería

Los instrumentos virtuales brindan significativas ventajas en cada etapa del proceso de ingeniería, desde la investigación y el diseño hasta el ensayo y puesta a punto del sistema final.

Diseño e investigación (I+D)

En la investigación y el diseño, se demandan capacidades de rápido desarrollo y realización de prototipos. Con los instrumentos virtuales se puede desarrollar rápidamente un programa, tomar mediciones desde un instrumento para ensayar un prototipo y analizar resultados, todo ello en un tiempo razonable. Cuando se necesita flexibilidad es esencial tener una plataforma ajustable y abierta, desde la computadora de mesa a los **sistemas embebidos** y redes distribuidas.

Los exigentes requerimientos de las aplicaciones de investigación y desarrollo (I+D) requiere una integración ininterrumpida de software y hardware. LabVIEW facilita la integración, tanto si se necesita comunicarse con instrumentos autónomos utilizando GPIB o adquirir de manera directa señales a la computadora con una placa de adquisición de datos y hardware de acondicionamiento de señales. Con los instrumentos virtuales también se puede automatizar un procedimiento de ensayo, minimizando la posibilidad de error humano y asegurando la consistencia de resultados al evitar introducir variables desconocidas o inesperadas.

1.17 Ensayos de desarrollo y validación

Dada la flexibilidad y poder de los instrumentos virtuales, se puede elaborar fácilmente complejos procedimientos de ensayo. En el campo de los ensayos automáticos de diseño de verificación, se crean rutinas en LabVIEW y posteriormente se integran con programas tales como el National Instruments TestStand, que ofrece capacidades de manejo de ensayos muy potentes. Además, si la ocasión lo requiere se puede desarrollar código dentro del proceso de diseño y luego insertar esos mismos programas dentro de herramientas funcionales.

1.18 Ensayo de planta

La obtención de menores tiempos de ensayo y la simplificación del desarrollo de procedimientos de ensayo son objetivos primarios en los ensayos de planta. Los instrumentos virtuales basados en LabVIEW, combinados con programas tales como el TestStand proveen alto rendimiento para colmar esas necesidades. Estas herramientas cumplen con requerimientos rigurosos de producción con una alta velocidad y un motor multitarea para correr múltiples secuencias de ensayo en paralelo. TestStand maneja fácilmente la secuencia de ensayo, ejecución y preparación de informes en base a rutinas escritas en LabVIEW. Así mismo, TestStand integra la creación de código de ensayo en LabVIEW y también puede reutilizar código creado en la etapa de I+D y la validación.

1.19 Aplicaciones en planta

Las aplicaciones en planta requieren que el software sea fiable, de alto rendimiento e interoperable. Los instrumentos virtuales basados en LabVIEW ofrecen todas estas ventajas mediante la integración de características tales

como: manejo de alarmas, tendencias de datos históricos, seguridad, redes, E/S industriales y conectividad empresarial.

Gracias a esta funcionalidad, se pueden conectar fácilmente muchas clases de equipos industriales, tales como PLCs, redes industriales, E/S distribuidas y placas de adquisición de datos insertables.

Compartiendo código a través de la empresa, la producción puede utilizar las mismas aplicaciones de LabVIEW desarrolladas en I+D o en la validación e integrarse de modo ininterrumpido con los procesos de ensayo de manufactura.

1.20 La Instrumentación Virtual más allá de la computadora personal

Recientemente, las tecnologías de las computadoras personales comerciales han comenzado a migrar hacia los **sistemas embebidos**. Ejemplos de ello incluyen Windows CE, procesadores Intel basados en X86, buses PCI y CompactPCI y Ethernet para el **desarrollo embebido**. Debido a que la instrumentación virtual se basa tan fuertemente en tecnologías comerciales para generar ventajas de costo y rendimiento, también se ha expandido para acompañar mayores **capacidades embebidas y de tiempo real**. Por ejemplo, LabVIEW corre sobre Linux así como también sobre el sistema operativo de tiempo real ETS de VenturCom sobre **objetivos embebidos específicos**. **La opción para utilizar la instrumentación virtual como un sistema de referencia ajustable que se extienda desde la computadora de mesa hasta los equipos embebidos debería considerarse una herramienta dentro de la caja de herramientas completa de un desarrollador de sistemas embebidos.**

Un ejemplo de cambio tecnológico que afecta el desarrollo de sistemas embebidos son las redes y la Web. Con la ubicuidad de las computadoras personales, Ethernet domina ahora como la infraestructura normal de redes de las empresas en todo el mundo. Además, la popularidad de la interface Web en el mundo de las computadoras personales se ha derramado sobre el desarrollo de teléfonos celulares, PSD y, actualmente, sistemas de adquisición de datos y control industriales.

Los **sistemas embebidos** alguna vez significaron operación autónoma o, como máximo, se comunicaron con componentes periféricos a bajo nivel a través de un bus de tiempo real. Ahora, la mayor demanda de información a todos los niveles empresarios (y en los productos de consumo) requieren que se instalen en red los sistemas embebidos.

Debido a que el software de instrumentación virtual puede combinar un medio ambiente de desarrollo tanto para computadoras de mesa como para sistemas de tiempo real utilizando tecnología combinada Inter-plataforma, se pueden capitalizar los servidores Web existentes y la facilidad de conexión a redes del software de los PCs y orientarlo **hacia los sistemas de tiempo real y embebidos**. Por ejemplo, se podría utilizar LabVIEW simplemente para configurar un servidor Web existente y exportar una interface de aplicación hacia máquinas seguras definidas sobre la red bajo Windows y luego descargar

esa aplicación para ejecutarla sobre un **sistema independiente embebido** que puede entrar en la palma de la mano del usuario. Este procedimiento sucede sin necesidad de programación adicional en el sistema embebido, posteriormente se puede distribuir ese sistema embebido, darle energía, conectarlo a la aplicación de una máquina remota segura via Ethernet y luego hacer la interface hacia él utilizando un servidor Web común. Para aplicaciones de redes más elaboradas, se programa gráficamente en TCP/IP u otros métodos con los cuales ya se ha familiarizado en LabVIEW y luego correrlos dentro del **sistema embebido**.

El desarrollo de **sistemas embebidos** es uno de los segmentos de la ingeniería de mayor crecimiento, y continuará siéndolo en el futuro cercano a medida que los consumidores demanden automóviles, artefactos del hogar, casas y demás productos con mayor inteligencia. La evolución de estas tecnologías comerciales impulsará la instrumentación virtual y la hará más utilizable en un creciente número de aplicaciones. Compañías líderes que proveen herramientas para el software y el hardware de instrumentación virtual necesitan invertir en experiencia y desarrollo de productos para servir este creciente conjunto de aplicaciones. Por ejemplo, para una plataforma de software de instrumentación virtual, se incluye la capacidad de ajustarse desde el desarrollo para sistemas operativos de computadoras de mesa hasta sistemas embebidos de tiempo real, a computadoras personales de mano, hardware basado en FPGAs, microcontroladores y aún permitir los sensores inteligentes (smart sensors).

La próxima generación de herramientas de instrumentación virtual necesita incluir tecnología de redes para lograr una rápida y fácil integración de Bluetooth, Ethernet inalámbrica y otras normas. Además de utilizar esas tecnologías, el software de instrumentación virtual requiere una mejor manera de describir y diseñar las relaciones de temporizado y sincronización entre sistemas distribuidos de una manera intuitiva para ayudar a lograr un más rápido desarrollo y control **de estos sistemas que son, a menudo, embebidos**. Los conceptos de software y hardware integrados de instrumentación virtual, herramientas modulares flexibles y la utilización de tecnologías comerciales se combinan para crear un marco de trabajo sobre el se puede completar rápidamente el desarrollo de sus sistemas y también mantenerlos en el largo plazo. Puesto que la instrumentación virtual ofrece tantas **opciones y capacidades en el desarrollo embebido** tiene sentido que los desarrolladores de sistemas embebidos analicen, comprendan y comprueben el campo operativo de esas herramientas.

1.22 Conclusión

La instrumentación virtual está dinamizada por la siempre creciente tecnología computacional que ofrece al usuario o equipo de trabajo el poder crear y definir su propio sistema basado en un marco de trabajo abierto. Este concepto no sólo asegura que el trabajo será utilizable en el futuro sino que también provee la flexibilidad de adaptarlo y extenderlo a medida que cambian las necesidades funcionales y operativas. LabVIEW fue diseñado teniendo en mente a los diseñadores de sistemas, provee herramientas poderosas y un entorno de

desarrollo familiar creado específicamente para el diseño de instrumentos virtuales.

2. SISTEMAS EMBEBIDOS

Un sistema embebido (también se lo conoce como incrustado ,embutido o integrado) se puede considerar como un sistema de computación con un propósito específico que está completamente encapsulado por el dispositivo que controla o monitoriza. Algunos ejemplos de sistemas embebidos son los sistemas de información integrados en automóviles, trenes o aviones, control de equipos, operación de maquinaria, y controladores de procesos en sistemas de producción industrial o plantas industriales completas.

El término embebido está caracterizando que esos circuitos integrados son una parte integral del sistema en que se encuentran. Lo interesante de que un sistema sea "embebido" es que puede estar de tal forma incrustado, puede quedar tan oculto a nuestros ojos, que la presencia de tales "chips" no resulte nada obvia a quien lo mira.

El uso de sistemas embebidos en productos complejos implica un desafío de la seguridad en las tecnologías de la información para proteger la información contenida en el sistema embebido y también la que es transmitida desde y hacia el dispositivo por redes privadas o Internet.

2.1 Componentes de un de un Sistema Embebido (SE)

En la parte central se encuentra el microprocesador, microcontrolador, DSP, etc.,es decir la CPU o unidad que aporta inteligencia al sistema. Según el sistema puede incluir memoria interna o externa, un micro con arquitectura específica según requisitos.

En general, un sistema embebido consiste en un sistema con microprocesador o microcontrolador cuyo hardware y software están específicamente diseñados y optimizados para resolver un problema concreto eficientemente.

Normalmente un sistema embebido interactúa continuamente con el entorno para vigilar o controlar algún proceso mediante una serie de sensores. Su hardware se diseña normalmente a nivel de chips, o de interconexión de PCBs, buscando la mínima circuitería y el menor tamaño para una aplicación particular.

Otra alternativa consiste en el diseño a nivel de PCBs consistente en el ensamblado de placas con microprocesadores comerciales que responden normalmente a un estándar como el PC-104 (placas de tamaño concreto que se interconectan entre sí "apilándolas" unas sobre otras, cada una de ellas con una funcionalidad específica dentro del objetivo global que tenga el sistema embebido). Esta última solución acelera el tiempo de diseño pero no optimiza ni el tamaño del sistema ni el número de componentes utilizados ni el coste unitario.

En general, un sistema embebido simple contará con un microprocesador o microcontrolador memoria, unos pocos periféricos de E/S y un programa dedicado a una aplicación concreta almacenado permanentemente en la memoria. El término embebido o empotrado hace referencia al hecho de que el microcomputador está encerrado o instalado dentro de un sistema mayor y su existencia como microcomputador puede no ser aparente.

Un usuario no técnico de un sistema embebido puede no ser consciente de que está usando un sistema computador. Por ejemplo en algunos hogares las personas, que no tienen por qué ser usuarias de un ordenador personal estándar (PC), utilizan del orden de diez o más sistemas embebidos cada día.

Los microcomputadores embebidos en estos sistemas controlan electrodomésticos tales como: televisores, videos, lavadoras, alarmas, teléfonos inalámbricos, etc. Incluso un PC tiene microcomputadores embebidos en el monitor, impresora, y periféricos en general, adicionales a la CPU del propio PC. Un automóvil puede tener hasta un centenar de microprocesadores y microcontroladores que controlan subsistemas como la ignición, transmisión, dirección asistida, frenos antibloqueo (ABS), control de la tracción, etc.

Los sistemas embebidos se caracterizan normalmente por la necesidad de dispositivos de E/S especiales. Cuando se opta por diseñar el sistema embebido partiendo de una placa con microcomputador también es necesario comprar o diseñar placas de E/S adicionales para cumplir con los requisitos de la aplicación concreta.

Muchos sistemas embebidos son sistemas de tiempo real. Un sistema de tiempo real debe responder, dentro de un intervalo restringido de tiempo, a eventos externos mediante la ejecución de la tarea asociada con cada evento. Los sistemas de tiempo real se pueden caracterizar como blandos o duros. Si un sistema de tiempo real blando no cumple con sus restricciones de tiempo, simplemente se degrada el rendimiento del sistema, pero si el sistema es de tiempo real duro y no cumple con sus restricciones de tiempo, el sistema fallará. Este fallo puede tener posiblemente consecuencias catastróficas.

Un sistema embebido complejo puede utilizar un sistema operativo como apoyo para la ejecución de sus programas, sobre todo cuando se requiere la ejecución simultánea de los mismos. Cuando se utiliza un sistema operativo lo más probable es que se trate de un sistema operativo en tiempo real (RTOS), que es un sistema operativo diseñado y optimizado para manejar fuertes restricciones de tiempo asociadas con eventos en aplicaciones de tiempo real. En una aplicación de tiempo real compleja la utilización de un RTOS multitarea puede simplificar el desarrollo del software.

La comunicación adquiere gran importancia en los sistemas embebidos. Lo normal es que el sistema pueda comunicarse mediante interfaces estándar de cable o inalámbricas. Así un sistema embebido normalmente incorporará puertos de comunicaciones del tipo RS232, RS485, SPI, I²C, CAN, USB, IP, WiFi, GSM, GPRS, DSRC, etc.

El subsistema de presentación tipo suele ser una pantalla gráfica, táctil, LCD, alfanumérico, etc. dependiendo del tipo de aplicación.

Denominamos actuadores a los posibles elementos electrónicos que el sistema se encarga de controlar. Puede ser un motor eléctrico, hidráulico, neumático, un conmutador tipo relé etc. El módulo de E/S analógicas y digitales suele emplearse para digitalizar señales analógicas procedentes de sensores, activar diodos LED, reconocer el estado abierto cerrado de un conmutador o pulsador, etc.

El módulo de alimentación se encarga de generar las diferentes tensiones y corrientes necesarias para alimentar los diferentes circuitos del sistema embebido. Usualmente se trabaja con un rango de tensiones de entrada que mediante convertidores AC/DC o DC/DC se obtienen las diferentes tensiones necesarias para alimentar los diversos componentes activos del circuito. Además de los convertidores AC/DC y DC/DC, otros módulos típicos, filtros, circuitos integrados supervisores de alimentación, etc. El consumo de energía puede ser determinante en el desarrollo de algunos sistemas embebidos que necesariamente se alimentan con baterías o energía fotovoltaica.

2.2 Ámbitos de aplicación de un sistema embebido.

Los lugares donde se pueden encontrar los sistemas embebidos son numerosos y de varias naturalezas. A continuación se exponen varios ejemplos para ilustrar las posibilidades de los mismos:

- I) En una fábrica, para controlar un proceso de montaje o producción. Una máquina que se encargue de una determinada tarea hoy en día contiene numerosos circuitos electrónicos y eléctricos para el control de motores, hornos, etc. que deben ser gobernados por un procesador, el cual ofrece un interfaz persona – máquina para ser dirigido por un operario e informarle al mismo de la marcha del proceso.
- II) Puntos de servicio o venta (POS, Point Of Service). Las cajas donde se paga la compra en un supermercado son cada vez más completas, integrando teclados numéricos, lectores de códigos de barras mediante láser, lectores de tarjetas bancarias de banda magnética o chip, pantalla alfanumérica de cristal líquido, etc. El sistema embebido en este caso requiere numerosos conectores de entrada y salida y unas características robustas para la operación continuada.
- III) Puntos de información al ciudadano. En oficinas de turismo, grandes almacenes, bibliotecas, etc. existen equipos con una pantalla táctil donde se puede pulsar sobre la misma y elegir la consulta a realizar, obteniendo una respuesta personalizada en un entorno gráfico amigable.
- IV) Decodificadores y set-top boxes para la recepción de televisión. Cada vez existe un mayor número de operadores de televisión que aprovechando las tecnologías vía satélite y de red de cable ofrecen un servicio de televisión de pago diferenciado del convencional. En primer lugar envían

la señal en formato digital MPEG-2 con lo que es necesario un procesado para decodificarla y mandarla al televisor. Además viaja cifrada para evitar que la reciban en abierto usuarios sin contrato, lo que requiere descifrarla en casa del abonado. También ofrecen un servicio de televisión interactiva o web-TV que necesita de un software específico para mostrar páginas web y con ello un sistema basado en procesador con salida de señal de televisión.

- V) Sistema radar. El procesado de la señal recibida o reflejada del sistema radar embarcado, sobre todo si se trata de un avión requiere de alta potencia de cálculo además de ocupar poco espacio, pesar poco y soportar condiciones extremas de funcionamiento (temperatura, presión atmosférica, vibraciones, etc
- VI) Equipos de electromedicina en hospitales y ambulancias UVI-MOVIL.
- VII) Máquinas de revelado de automático de fotos.
- VIII) Cajeros automáticos.
- IX) Pasarelas (Gateways) Internet-LAN.
- X) Y un sinfín de posibilidades aún por descubrir o en estado embrionario como pueden ser los PCs de bolsillo.

2.3 Ventajas de un sistema embebido sobre las soluciones industriales tradicionales

Los equipos industriales de medida y control tradicionales están basados en un microprocesador con un sistema operativo propietario o específico para la aplicación correspondiente. Dicha aplicación se programa en ensamblador para el microprocesador dado o en lenguaje C, realizando llamadas a las funciones básicas de ese sistema operativo que en ciertos casos ni siquiera llega a existir. Con los modernos sistemas PC embebido basados en microprocesadores i486 o i586 se llega a integrar el mundo del PC compatible con las aplicaciones industriales. Ello implica numerosas ventajas:

- Posibilidad de utilización de sistemas operativos potentes que ya realizan numerosas tareas: comunicaciones por redes de datos, soporte gráfico, concurrencia con lanzamiento de threads, etc. Estos sistemas operativos pueden ser los mismos que para PCs compatibles (Linux, Windows, MS-DOS) con fuertes exigencias en hardware o bien ser una versión reducida de los mismos con características orientadas a los **PCs embebidos**.
- Al utilizar dichos sistemas operativos se pueden encontrar fácilmente herramientas de desarrollo software potentes así como numerosos programadores que las dominan, dada la extensión mundial de las aplicaciones para PCs compatibles.
- Reducción en el precio de los componentes hardware y software debido a la gran disponibilidad de PCs.

- El apelativo de **Sistema Embebido** (embedded) refleja que son una parte integral (interna) del sistema y en general son dispositivos utilizados para controlar o asistir la operatividad de equipamiento muy diverso.

La importancia actual de los sistema embebidos es de tal magnitud que ya en el 6º Programa Marco sobre Tecnologías para la Sociedad de la Información se hace hincapié en la necesidad de investigar nuevos horizontes operativos y funcionales sobre la tecnología de dichos sistemas.



Objetivos Estratégicos

- **Reforzar el liderazgo en áreas de fortaleza**
 - comunicaciones y servicios móviles, comunicaciones de banda ancha
 - micro-opto electrónica & μ sistemas,
 - sistemas audiovisuales en red...
- **Explorar nuevas oportunidades de la visión del Ambiente Inteligente y eliminar debilidades para su realización**
 - Sistemas embebidos, integración de dispositivos sensores y actuadores
 - Interfases multimodo, gestión semántica del conocimiento, contenido
 - Seguridad y dependencia, estandars y herramientas abiertos
 - Redes de conocimiento y computación

SGOPIGI - SGPC - SEPOCYT

2.4 Sistemas embebidos tradicionales.

Los microcontroladores, DSPs y FPGAs son los elementos mayoritariamente empleados en el diseño e implementación de los sistemas embebidos tradicionales, aunque existen aplicaciones asociadas al diseño de sistemas electrónicos embebidos reconfigurables que emplean SOC's (System On Chips) Una tarea típica de los microcontroladores es el control mediante eventos externos. El estado del entorno exterior es detectado mediante las entradas A/D y líneas de I/O. Cada entrada contiene información relativa al estado de una parte muy específica del sistema, mientras que las salidas se dirigen hacia los actuadores que modifican la operatividad del sistema.

Otra aplicación, las comunicaciones embebidas que aportan una nueva tecnología orientada a la conectividad a través de Internet. En la cabecera se encuentra una DSP mientras que en el nivel más bajo se dispone de microcontroladores de 8/16/32 bits dependiendo del tipo de sistema. En general las DSPs se emplean donde sea necesario capacidad de procesamiento de señales y cálculos numéricos.

En otras ocasiones los microcontroladores suelen emplearse en lugar de los ASICS debido a su menor coste y mayor flexibilidad. Aunque tienen una capacidad de cálculo inferior a una DSP, se emplean mayoritariamente en aplicaciones de gran volumen y bajo costo como por ejemplo en los sistemas de inyección de un motor.

Las FPGAs se aplican sobre todo en aquellos subsistemas donde el requisito fundamental es la velocidad.

Un aspecto básico es el relativo a los algoritmos, arquitecturas y herramientas para el diseño e implementación de los sistemas empotrados.

Por una parte se considera el análisis algorítmico con toda la problemática asociada a los lenguajes de especificación. Por otra parte, se evalúa el diseño arquitectural considerando las posibles restricciones temporales, de coste o de consumo que son habituales en el diseño de estos sistemas. Por último, también es posible abordar el desarrollo de herramientas ad-hoc para resolver determinadas tareas de diseño persiguiendo soluciones que contemplen el mayor grado de optimización posible. De una forma generalista las distintas fases correspondientes al desarrollo de una metodología aplicada al ámbito industrial para el diseño e implementación de sistemas embebidos corresponden a:

- Estudio bibliográfico para determinar las especificaciones de un sistema empotrado concreto. Se plantean ejemplos concretos de sistemas empotrados, siendo de especial interés la implementación de sistemas para la codificación de audio, vídeo y la transmisión de datos IP.
 - Análisis algorítmico. Evaluación de los bloques fundamentales desde el punto de vista microelectrónico para el posterior diseño arquitectural. Esta evaluación pretende determinar aquellos bloques que sean más exigentes de acuerdo con los requisitos del sistema (coste, tiempos, consumo, etc.)
 - Especificación del sistema. Modelado mediante un lenguaje concreto (Verilog, VHDL) para los subsistemas implementados en FPGAs y en C para los subsistemas implementados en un microcontrolador o DSP. También se utilizan a nivel global otros lenguajes como C++ y Java.
 - Estudio bibliográfico sobre las posibles técnicas disponibles para mejorar el módulo y aspecto elegido en la fase anterior.
 - Selección de las herramientas y la tecnología destino. Dependiendo del problema puede ser necesario el desarrollo de una herramienta específica dentro del marco del trabajo a desarrollar.
 - Según el análisis del punto anterior, especificación y desarrollo de la herramienta de ayuda (CAD) que se haya determinado como necesaria para alcanzar los objetivos planteados para el diseño. Se incluye el estudio de la bibliografía disponible, la selección del entorno de programación a utilizar y la integración final de la herramienta en el proceso de diseño.
-

- Implementación del sistema empotrado y evaluación del mismo mediante benchmarks consistentes con la especificación inicial.

Cuando el acento de la aplicación recae en microcontrolador es posible incluso dotarle de un pequeño sistema operativo ROM-DOS que se puede considerar como el DOS de los sistemas embebidos. <http://www.rom-dos.com>

- Además de su reducido tamaño y de los escasos recursos que necesita, ROM-DOS posee algunas características que pueden marcar la diferencia de forma significativa:
- RXE (Relocatable Executable): En DOS, los programas se cargan en RAM antes de proceder a su ejecución. La tecnología RXE permite convertir un programa estándar en un programa que se puede ejecutar directamente desde ROM.
- ROM-DOS BUILD Utility: Permite añadir y quitar características al sistema operativo, reduciendo así el tamaño final del kernel.
- Carga dinámica de drivers de dispositivo: ROM-DOSTM tiene la capacidad de detectar el hardware y cargar drivers de dispositivo de forma dinámica, lo que le permite autoconfigurarse según el entorno.

Tarjetas que implementan subsistemas embebidos basados en microcontroladores, DSPS y FPGAS las podemos encontrar en:

Equipos de medida de Parámetros Mecánicos. Medidas angulares. Medidas de fatiga. Medidas de distancia. Agujeros. Medidas peliculares.

Equipos de medida de EMI. Amplificadores de potencia. Antenas. Receptores de medida.

Instrumentación Virtual. Plataformas de prueba para automoción. Aplicaciones en tiempo real.

Sistemas automáticos de inspección y reconocimiento visual. Posición de componentes.

Protección ESD. Entorno de trabajo. Equipamiento individual. Clases de pavimentos. Productos de limpieza. Equipamiento de las estaciones de trabajo. Embalajes.

Equipos para medidas ópticas. Generadores de señal. Atenuadores. Medidores de potencia. Analizadores de espectro.

Equipos para la medida de señales digitales. Generadores lógicos. Generadores de palabras. Tester lógicos. Analizadores lógicos.

Equipamiento de medida de propósito general. Medidores de capacidad. Contadores. Generadores de funciones. Medidores de impedancia. Medidores de Inductancia. Multímetros.

Equipos para la medida de EMI. Dispositivos para la medida de Intensidad de Campo. Medidores de campo magnético. Equipamiento EMG para pruebas de inmunidad. Comprobadores de ESD.

Equipos para la medida de RF. Frecuencímetros. Receptores de medida. Analizadores de redes Escalares y Vectoriales. Medidores de la figura de ruido. Generadores de ruido. Medidores de potencia.

Equipos de medida para Comunicaciones.- Analizadores RDSI. Analizadores ATM. Analizadores LAN y WLAN.

Equipamiento de medida de propósito general. Osciloscopios. Medidores de Potencia. Generadores de pulsos. Medidores de Resistencia. Generadores de señal. Medidores de transitorios.

Equipos de medida de radio móvil. Sistemas de prueba para equipos analógicos y digitales. Generadores de señal. Analizadores de señal digital. Sistemas de análisis de cobertura.

Nanotecnología. Micro y nanoensamblaje. Componentes. Técnicas de interconexión. Microrobótica. Nanorobótica. Tecnologías de prueba en microsistemas.

Equipos para la medida de micro-ondas. Accesorios para la medida de micro-ondas. Medida de cables RF. Puentes de medida. Kits de calibración.

Motores eléctricos.- Alterna. Continua. Motores de Par. Paso a paso. Sin escobillas. Micromotores. Controladores lineales. Controladores de motores de pasos. Actuadores magnéticos.

Equipos para la medida de RF.- Reflectómetros. Generadores de señal. analizadores de espectros. Generadores de barrido.

Montaje automático de componentes.- Equipos para montaje en superficie. Equipos para inserción en placa. Montaje semiautomático. Máquinas pick and place para componentes especiales. Máquinas de inserción de pines. Montaje de BGAs. Sistemas de ensamblaje de Flip Chip.

Alimentación.- Componentes. Arquitectura de potencia distribuida.

Equipos para medidas ópticas.- Analizadores de redes. Analizadores de polarización. Amplificadores. Multímetros. Medidas de dispersión en FO. Medidores de intensidad luminosa.

Pruebas en Comunicaciones.- Analizadores WAN. Comprobadores de señalización nº 7. Equipos de prueba PCM/PDH. Equipos de medida para DTV y Radio Digital.

Cajas para pared y sobremesa.- Estructuras. Consolas. Cajas para ordenador. De sobremesa. Metálicas.

Sistemas de prueba de Buses.- CAN. Field-bus. I2C. PCI. USB.

Componentes para la medida de micro-ondas.- Atenuadores. Acopladores. Mezcladores. Divisores de potencia.

Medidas de seguridad.- Medida de puestas a tierra. Comprobadores de aislamiento. Localización de cortocircuitos. Sistemas de prueba de estándares de seguridad (VDE, UL, TUV.). Comprobadores de AT inductivos.

Instrumentación Virtual.- Módulos I/O para aplicaciones industriales.

Medida de parámetros mecánicos.- Fuerza. Presión. Par. Medidores de vacío. Medidores de viscosidad.

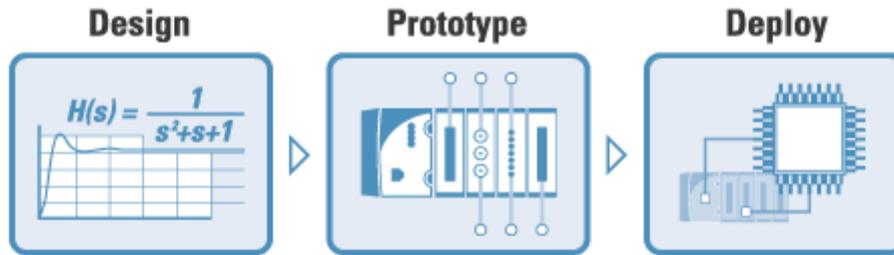
Fuentes de alimentación de potencia.- Fuentes de alimentación. Convertidores AC/DC. Convertidores DC/DC.

3. LABVIEW PARA SISTEMAS EMBEBIDOS.

Como ha referenciado anteriormente, el diseño e implementación de un sistema embebido puede ser una tarea bastante compleja. Efectuar una completa y exitosa integración entre hardware, y software requiere de complejas y sofisticadas herramientas de modelización y simulación. Es común en los desarrolladores de software construir modelos basados en C, C++ y Java para posteriormente ser testeados y probados en un ordenador (host computer) antes de escoger el hardware final (target hardware basado en microcontroladores, DSPS y FPGAS), es decir desde el análisis del problema a su implementación., <http://community.ni.com/embedded>.

Los módulos que dispone la plataforma LabVIEW para el desarrollo de sistemas embebidos son:

-
- **NI LabVIEW**
 - **LabVIEW Development Systems**
 - **LabVIEW PDA Module**
 - **LabVIEW Real-Time**
 - **LabVIEW for Embedded Applications**
 - **NI LabVIEW Embedded Module for ADI Blackfin Processors**
 - **LabVIEW FPGA Module**
 - **LabVIEW Embedded Development Module**
 - **LabVIEW DSP Module**
 - **CompactRIO**
 - **CompactField Point**
 - **Links**
 - **LabVIEW Datalogging and Supervisory Control Module**
 - **LabVIEW Touch Panel Module**
 - **LabVIEW for Machine Vision**



Learn How to Design Faster through Webcasts by Analog Devices, Celoxica, and Maplesoft. >>

3.1 LabVIEW Real-Time

Los sistemas embebidos se enfrentan normalmente a problemas de tiempo real. En la actualidad se pueden desarrollar aplicaciones en tiempo real y con menor coste sin necesidad de poseer amplios conocimientos de programación en lenguajes tradicionales basados en texto, usando el software LabVIEW RT y las herramientas compatibles con este entorno.

Para muchas aplicaciones, una gran parte del coste de los sistemas de medida y automatización reside en el desarrollo de las **aplicaciones de control en tiempo real y de los sistemas embebidos**, pese a que las herramientas necesarias para el desarrollo de estas aplicaciones constituyen tan sólo una pequeña fracción del sistema completo. El resultado es un coste elevado, ya que el desarrollo de sistemas en tiempo real requiere típicamente conocimientos de programación a nivel de registros y la instalación de hardware a medida. Las soluciones personalizadas pueden presentar problemas porque resultan difíciles de mantener y de integrar con otros sistemas, y no se adaptan con facilidad al empleo de nuevas tecnologías. El uso de tecnología PC estándar puede resolver estos problemas, disminuir los costes y mejorar la integración del software y del hardware. Hasta ahora esto era difícil. Ejecutar aplicaciones de control en tiempo real en un entorno Windows tradicional conlleva tiempos de respuesta inciertos, porque Windows es un sistema operativo no determinista. En consecuencia, los tiempos de ciclo del bucle de control pueden variar de una iteración a la siguiente.

3.2 LabVIEW RT combina flexibilidad de programación y facilidad de uso.

La programación gráfica en LabVIEW, combinada con su estrecha integración a un amplio abanico de dispositivos hardware, es ideal para dar el salto de la medición y automatización basadas en ordenador al desarrollo de **sistemas en tiempo real y sistemas embebidos**. La simplicidad de LabVIEW oculta un potente compilador multiproceso de 32 bits y un lenguaje de programación gráfica que permite a los ingenieros resolver los problemas más complejos de medición y automatización.

Con LabVIEW RT, se pueden desarrollar sus aplicaciones mediante la programación gráfica en el entorno familiar de Windows. Funciona exactamente igual que el software LabVIEW de programación gráfica. (VIs, Virtual

Instruments) de DAQ y de procesamiento de señal, así como otras funciones y estructuras usadas en LabVIEW se utilizan también en LabVIEW RT. La diferencia fundamental es que el código generado con este último sistema se puede descargar para ser ejecutado sobre hardware independiente de adquisición de datos en tiempo real.

3.3 NI LabVIEW Embedded Module for ADI Blackfin Processors

Entre las características funcionales más significativas están:

- Ofrece la capacidad para reutilizar algoritmos embebidos ya existentes o software embebido desarrollado previamente.
- Incluye controladores de componentes completamente integrados para convertidores ADI
- Se enfoca en la familia de procesadores embebidos Blackfin de baja potencia y alto rendimiento
- Proporciona una solución completamente integrada desde la formación del concepto hasta el despliegue

En la figura 4 se muestra la estructura jerárquica correspondiente a las distintas fases de diseño con este módulo.

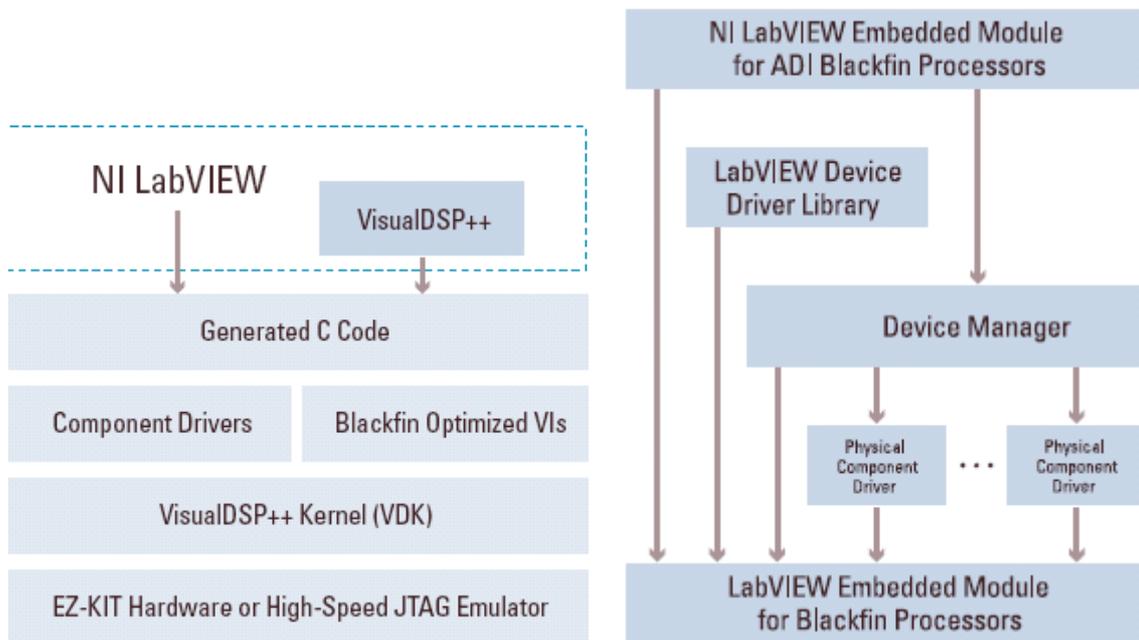


Fig. 4 . Estructura jerárquica de diseño

Este módulo LabVIEW Embedded Module for Blackfin Processors <http://ni.com/labview/blackfin> es compatible con todos los emuladores JTAG de Analog Devices y se dispone de los siguientes modelos

- NI LabVIEW Embedded Module for ADI Blackfin Processors Bundle

- NI LabVIEW Embedded Module for ADI Blackfin Processors Bundle SSP
- NI LabVIEW Embedded Module for ADI Blackfin Processors
- NI LabVIEW Embedded Module for ADI Blackfin Processors with SSP
- SSP, LabVIEW Embedded Module for Blackfin Processors

3.4 LabVIEW FPGA Module

La plataforma LabVIEW y el Módulo LabVIEW FPGA proporcionan un entorno de desarrollo gráfico sistemas digitales basados en (FPGAs). Con este módulo LabVIEW FPGA, se pueden desarrollar VIs FPGA en un servidor ejecutando Windows y LabVIEW compila e implementa el código de hardware. De esta forma se pueden desarrollar VIs de FPGA embebidos que combinan acceso directo a E/S con la lógica de LabVIEW definida por el usuario para definir hardware personalizado para aplicaciones como protocolos de comunicación digital, simulación de control en hardware y rápida generación de prototipos de control.

El Sistema de Desarrollo de Software de E/S Reconfigurable de LabVIEW incluye todas las herramientas que son necesarias para implementar un sistema embebido CompactRIO -- LabVIEW FPGA para sintetizar hardware personalizado en el FPGA de E/S reconfigurable, configurado por el usuario y LabVIEW Real-Time para crear aplicaciones determinísticas en tiempo real.

A modo de resumen las características operativas más notables son:

- Extiende la plataforma de LabVIEW a FPGAs
- Requiere cero conocimiento de VHDL o herramientas de diseño basadas en lenguajes de descripción del hardware.
- Brinda resolución personalizada de temporización y disparo de 25 ns y velocidades de hasta 80 MHz
- Diseña inteligencia en plataformas comerciales de hardware

3.5 LabVIEW Embedded Development Module

Este módulo permite a los ingenieros de diseño utilizar el entorno gráfico de programación de LabVIEW el diseño, simulación, implementación y desarrollo de algoritmos para sistemas embebidos utilizando procesadores y microcontroladores de 32 bits.

Evidentemente, la manera más eficiente de implementar un programa en un procesador de cualquier clase es utilizar el lenguaje ensamblador de dicho procesador. Sin embargo con frecuencia los diseñadores prefieren sacrificar un poco la eficiencia del código en búsqueda de facilitar y acortar el proceso de diseño.

En la actualidad, gran parte de los diseñadores utilizan lenguaje C en el desarrollo de sus programas. La universalidad de este programa facilita la implementación de los diseños en dispositivos de distintos fabricantes,

adicionalmente los avances en los compiladores actuales permiten lograr eficiencias de código de alrededor de un 90%.

Sin embargo, para aquellos usuarios que no estén muy familiarizados con la programación de software embebido, o para aplicaciones donde el tiempo de desarrollo sea el factor determinante en la implementación de su sistema embebido, utilizar técnicas de programación gráfica es una opción bastante atractiva. Es precisamente a estos usuarios a los que está dirigido el nuevo módulo de LabVIEW.

LabVIEW permite a los diseñadores de sistemas embebidos desarrollar sus sistemas embebidos en un ambiente gráfico de programación, y después traducir el código a lenguaje C o implementarlo directamente en el dispositivo, utilizando las herramientas apropiadas.

El módulo de sistemas embebidos incluye más de 400 funciones de análisis para procesamiento de señales, algebra lineal, ajuste de curvas, estadísticas y cálculos, además de incluir algunos indicadores y controles que permite a los diseñadores desarrollar, probar y depurar su código de manera interactiva. En la figura 5 se muestra el proceso de desarrollo.

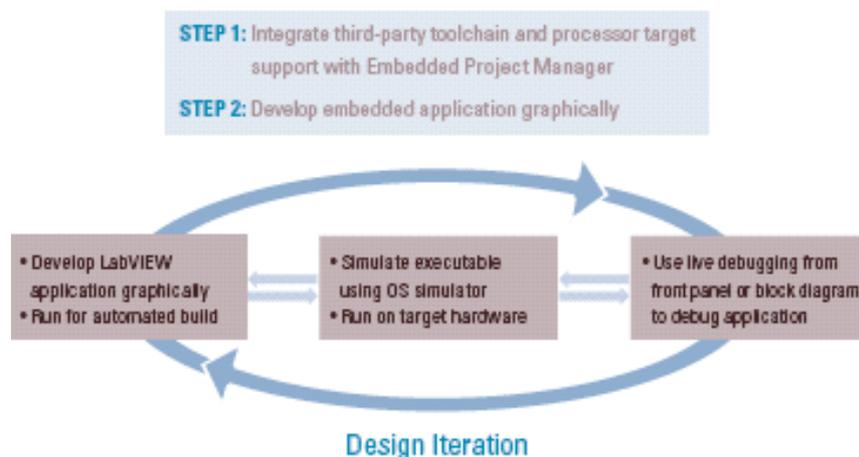


Fig. 5 .Diseño de una aplicación

3.6 LabVIEW DSP Module

Este módulo LabVIEW DSP permite un rápido desarrollo de aplicaciones con DSP. Soporta la programación gráfica de varias tarjetas de DSPs incluyendo la NI SPEEDY-33, Texas Instruments TMS320C6713 DSK y TMS320C6711 DSK. Para la programación gráfica de la tarjeta elegida se puede hacer uso de todas las funciones y librerías de la plataforma LabVIEW correspondientes al dominio del tiempo, de la frecuencia, transformadas, gestión de ficheros, etc. Entre los ítems funcionales más notables cabe referenciar:

- Necesita tener los Sistemas de Desarrollo de LabVIEW Profesional o Completo.
- Cientos de funciones DSP integradas para un desarrollo más rápido.
- Capacidad autónoma de desarrollo de aplicaciones.

- Express VIs facilitan el desarrollo y acortan los tiempos de desarrollo.
- Soporte para tres objetivos DSP que proporciona opciones para la implementación final
- Implementación de filtros digitales diseñados usando el tool de herramientas para Diseño de Filtros Digitales

En la figura 6 se muestra la implementación de un sistema heterodino y en la figura 7 se muestra le panel frontal (interface de usuario GUI) y le diagrama de bloques (programación gráfica) de una aplicación empleando el módulo NI LabVIEW DSP para NI SPEEDY-33 Y TI DSKs.

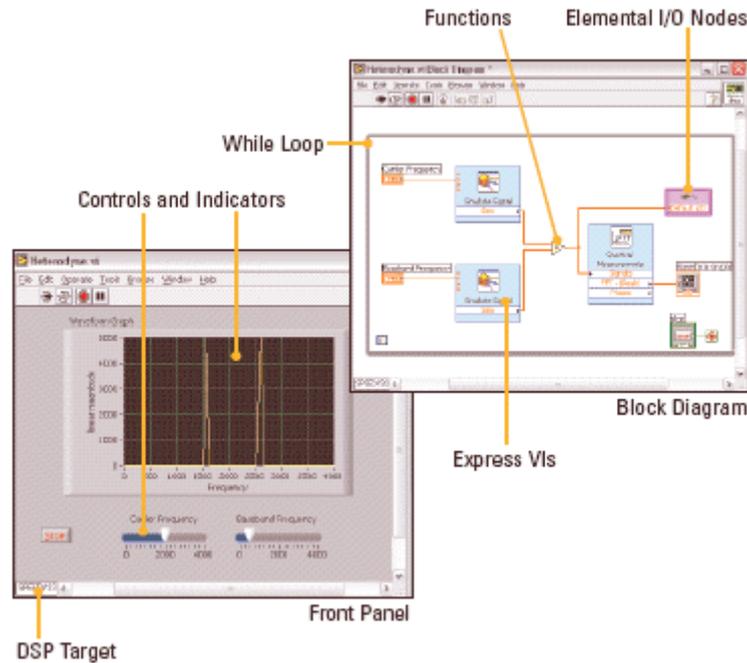


Fig. 6. Implementación de un sistema heterodino empleando LabVIEW DSP

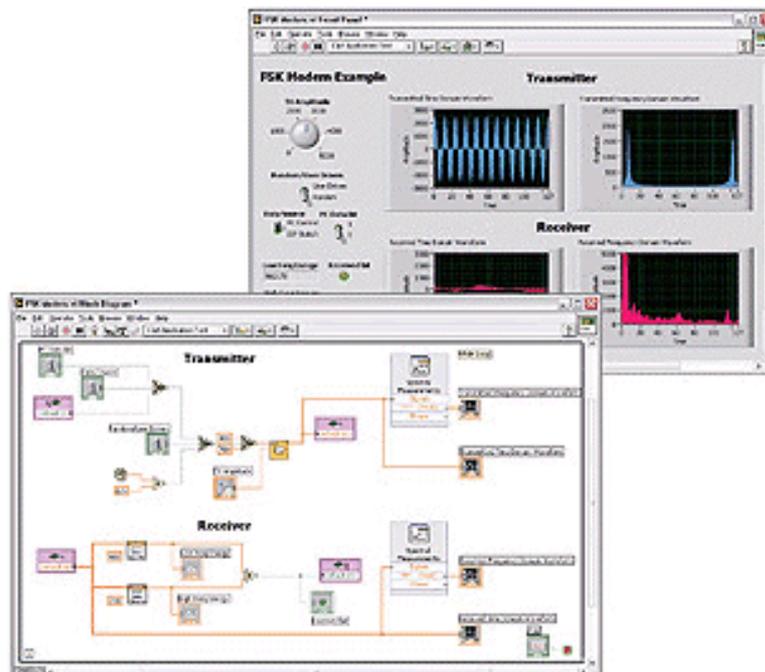


Fig.7. Programación para una tarjeta NI SPEEDY-33 Y TI DSKs.

Como resumen mediante **LabVIEW para Aplicaciones Embebidas** es posible **conectar objetivos embebidos** como microprocesadores de 32 bit, FPGAs y DSPs con el ambiente de desarrollo gráfico de LabVIEW de National Instruments. Recordar que con el Módulo de Desarrollo Embebido de LabVIEW, se puede programar gráficamente cualquier microprocesador de 32-bit. El Módulo LabVIEW FPGA proporciona un ambiente de programación gráfica en campo para FPGAs y además integrarlos en objetivos de hardware de E/S (RIO) reconfigurables, aspecto que se tratará en el siguiente apartado . Con estos productos, se combina el acceso directo a E/S con la lógica de LabVIEW definida por el usuario para definir hardware personalizado. En resumen:

- Aprovechar la depuración interactiva basada en el panel frontal y el diagrama de bloques.
- Programar gráficamente microprocesadores y microcontroladores de 32 bits.
- Configurar gráficamente FPGAs en objetivos de hardware de E/S reconfigurables.
- Desarrollar aplicaciones para microprocesadores de 32 bit , FPGAs y DSPs.
- Ejecutar simultáneamente múltiples tareas.

3.7 CompactRIO. Sistema Reconfigurable de Control y Adquisición de Datos

El CompactRIO de National Instruments es un sistema industrial de control y adquisición de datos pequeño y robusto que está motorizado por tecnología reconfigurable de E/S del tipo FPGA (RIO) para proveer un alto rendimiento y permitir la personalización. El NI CompactRIO incorpora un procesador FPGA de tiempo real y reconfigurable para ejecutar aplicaciones fiables e independientes o distribuidas y módulos industriales de E/S removibles en caliente que poseen acondicionamiento de señal incorporado para poder conectar directamente sensores y actuadores, figura 8

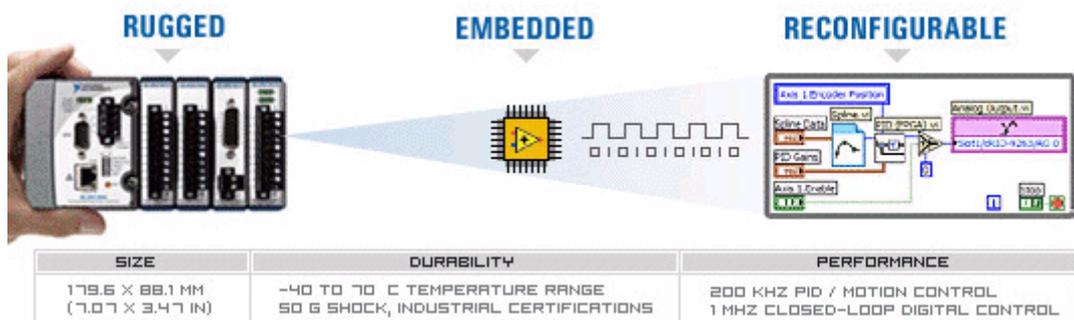


Fig.8. CompactRIO. Sistema Reconfigurable de Control y Adquisición de Datos

CompactRIO representa una arquitectura de bajo costo con acceso abierto a los recursos de bajo nivel del hardware. Sus **sistemas embebidos** se desarrollaron utilizando herramientas gráficas de programación en LabVIEW para un rápido desarrollo. Con CompactRIO se pueden implementar rápidamente sistemas embebidos de control o adquisición que rivalizan con el rendimiento y optimización de los circuitos de hardware diseñados a medida. La plataforma Compact RIO incluye los controladores de tiempo real cRIO-9002 y cRIO-9004 con procesadores industriales de punto flotante, la familia cRIO-910x de chasis reconfigurables de 4 y 8 ranuras equipados con FPGAs de 1 o 3 millones de compuertas, así como también una amplia variedad de tipos de E/S, desde entradas de termopares de ± 80 mV hasta entradas digitales universales de 250 VCA/VCC.

Los **sistemas embebidos** CompactRIO se desarrollan utilizando LabVIEW. Un sistema embebido CompactRIO tiene incorporado un procesador de punto flotante, un chasis reconfigurable de 4 y 8 ranuras que contiene un FPGA programable por el usuario y módulos industriales de E/S removibles en caliente.

Esta **arquitectura embebida** de bajo costo permite un acceso abierto a los recursos de bajo nivel del hardware para lograr un rápido desarrollo de sistemas independientes hechos a medida o de sistemas distribuidos de adquisición y control.

El sistema de expansión CompactRIO Serie R utiliza los mismos módulos industriales de E/S removibles en caliente para proveer un acondicionamiento de señales de alto rendimiento y expansiones de E/S para dispositivos FGPA PCI o PXI/ CompactPCI Serie R. El sistema de expansión agrega capacidades de mediciones personalizadas a las aplicaciones, tales como dispositivos de adquisición tradicional mediante placas insertables, visión, movimiento de maquinaria e instrumentación modular.

3.7.1 Arquitectura Abierta de Bajo Costo

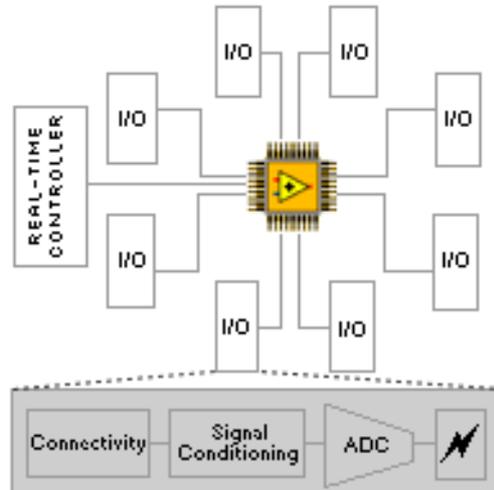
El CompactRIO combina un procesador embebido de tiempo real de bajo consumo con un chip RIO FPGA de alto rendimiento.

El núcleo RIO posee mecanismos propios de transferencia de datos para transferirlos al procesador embebido a fin de realizar análisis en tiempo real, post-procesamiento, *data-logging* o comunicación a una computadora servidor en la red.

CompactRIO provee un acceso directo al hardware desde los circuitos de entrada y salida de cada módulo de E/S utilizando funciones elementales de E/S de LabVIEW FPGA. Cada módulo de E/S posee **conectividad embebida**, acondicionamiento de señal, circuitos de conversión (tales como ADC o DAC) y una barrera opcional optoacoplada. Este diseño representa una arquitectura de bajo costo con acceso abierto a los recursos de bajo nivel del hardware.

3.7.2 Módulos de E/S

Cada módulo de E/S CompactRIO contiene incorporado un acondicionador de señales. Hay disponible una variedad de módulos de E/S, incluyendo entradas de termopares de ± 80 mV, entradas y salidas analógicas con muestreo simultáneo, E/S digitales industriales de 24 V de hasta 1 mA, entradas digitales diferenciales/TTL con salida de voltaje regulada de 5 V para codificadores (*encoders*) y entradas digitales universales de 250 Vrms. Debido a que los módulos contienen acondicionamiento de señales para rangos extendidos de voltajes o señales de tipo industrial, usualmente se pueden cablear las conexiones directamente desde el módulo



3.7.3 Procesador de Tiempo Real

El **sistema embebido** CompactRIO posee un procesador industrial del tipo Pentium 200 MHz que ejecuta de manera confiable y determinística las aplicaciones LabVIEW Real-Time. Se puede elegir entre miles de funciones disponibles en LabVIEW para **construir un sistema embebido** multihilo para control en tiempo real, análisis, *data-logging* y comunicaciones. El controlador también posee un puerto Ethernet 10/100 para una comunicación programable sobre la red (incluyendo correo electrónico) y servidores Web (HTTP) y de archivo (FTP).

3.7.4 Chasis Reconfigurable

El chasis reconfigurable es el corazón de los sistemas embebidos NI CompactRIO y contienen el núcleo RIO FPGA. Este RIO FPGA configurable por el usuario es una implementación a medida del diseño de la lógica de control, entradas/salidas, temporización, disparos y sincronización.

3.7.5 Field-Programmable Gate Arrays (FPGAs)

Debido a su rendimiento, posibilidad de reconfiguración, pequeño tamaño y bajos costos de desarrollo de ingeniería, los dispositivos FPGA son ampliamente utilizados por los vendedores de sistemas de adquisición y control. Tradicionalmente, estos dispositivos han sido definidos por los fabricantes en lugar de ser definidos por el usuario debido a la complejidad de

las herramientas de diseño electrónico. Ahora se puede aprovechar los FPGAs programables por el usuario para crear sistemas de adquisición y control reconfigurables altamente optimizados sin la necesidad de conocer lenguajes especializados de diseño de hardware, tales como el VHDL. Con el CompactRIO se pueden diseñar circuitos de adquisición y control personalizados en silicio con una resolución de temporizado/disparo de 25 ns. Los dispositivos FPGA poseen una arquitectura digital reconfigurable con una matriz de bloques lógicos configurables (CLBs) rodeados por una periferia de bloques de E/S. Las señales pueden ser dirigidas dentro de la matriz FPGA de una manera arbitraria mediante llaves (*switches*) programables interconectados y rutas cableadas. E

3.7.6 Rendimiento

Utilizando el software LabVIEW FPGA y la tecnología de hardware reconfigurable, se pueden crear sistemas de adquisición y control de máximo rendimiento con CompactRIO. El circuito FPGA es una máquina de cálculo reconfigurable de procesamiento paralelo que ejecuta la aplicación LabVIEW en circuito de silicio sobre un chip. Se pueden diseñar circuitos de adquisición y control personalizados sobre silicio con una resolución de 25 ns en el temporizado/disparo. El módulo LabVIEW FPGA posee **funciones embebidas** para efectuar lazos cerrados de control PID, filtros FIR de quinto orden, tablas *look-up* unidimensionales, interpolación lineal, detección de cruces por el cero y sintetización directa de señales digitales senoidales.

Con el **hardware embebido RIO FPGA** se pueden implementar sistemas de control PID analógicos multilazo con velocidades que exceden los 100.000 muestras/s. Se pueden también implementar sistemas digitales de control con velocidades de lazo de hasta 1.000.000 muestras/ s y es posible evaluar múltiples ciclos Booleanos utilizando lazos *while* simples de hasta 40 MHz (25 ns). Debido a la naturaleza paralela del núcleo RIO, agregar cálculos adicionales no reduce necesariamente la velocidad de la aplicación FPGA.

3.7.7 Diseño de Aplicaciones de Control en Tiempo Real

Los sistemas reconfigurables de control y adquisición de datos contienen normalmente cuatro componentes principales:

- Aplicación central RIO FPGA para entradas, salidas, comunicaciones y control.
- Lazo de tiempo crítico para control de punto flotante, procesamiento de señales, análisis y toma de decisiones punto a punto.
- Lazo de prioridad normal para *data logging* embebido, interface de panel remoto Web y comunicación Ethernet/ serie.
- PC Servidora en red para interface gráfica del usuario de modo remoto, *Data Logging* histórico y postprocesamiento. Dependiendo de los requerimientos de la aplicación, se puede decidir la implementación de alguno o todos los componentes.

.8 CompactField Point. Control Embebido

Cuando es necesario implementar un sistema de control embebido o realizar medidas en un espacio reducido, los módulos asociados al entorno E/S Compact FieldPoint es una opción vanguardista, robusta, precisa y fiable. Desde la producción de los módulos controladores FieldPoint FP-2000, los desarrolladores de LabVIEW han sido capaces de crear aplicaciones y desplegarlas en controladores Compact FieldPoint.

3.8.1 Encogiendo a LabVIEW

Todos los controladores pequeños de Compact FieldPoint ejecutan LabVIEW Real-Time. Con LabVIEW Real-Time, se puede usar la paleta de funciones estándar de LabVIEW para realizar análisis de datos, registro y almacenamiento de datos y comunicación por red. El usuario desarrolla y depura su sistema en Windows y luego descarga y ejecuta su programa en el procesador dedicado en el controlador Compact FieldPoint.

Compact FieldPoint es capaz de desempeñar ciclos de control a tasas de hasta 200 veces/seg. Al distribuir LabVIEW justo al lado de sus fuentes de señal, se **elimina el PC**, lo cual incrementa la comodidad de su sistema, reduce la complejidad del cableado y disminuye el riesgo de captar ruido en cableados largos. Cada módulo del entorno contiene memoria Flash interna no volátil para **almacenar aplicaciones embebidas** y almacenar datos. También contienen memoria DRAM utilizada para ejecutar las aplicaciones de LabVIEW embebidas.. Todos los controladores ejecutan un sistema operativo a tiempo real (RTOS), NO basado en Windows, con bajos requisitos de procesamiento en un procesador x86 Intel.

- El sistema operativo a tiempo real que se ejecuta en cada módulo proporciona dos ventajas sobre un sistema operativo típico.
- Este sistema operativo no tiene la complejidad de un sistema operativo estándar. Esto significa que puede operar con menos recursos. El sistema operativo y la máquina de LabVIEW Real-Time 6.1 utiliza 8 MB de DRAM y 6 MB de Flash. El código de LabVIEW se ejecuta rápida y eficazmente en una arquitectura con un procesador x86 de bajo consumo. Debido a que el sistema operativo esta racionalizado, no sufre de las inestabilidades inherentes a un sistema operativo complejo.
- El sistema operativo a tiempo real tiene la capacidad de asignar prioridades a cada ciclo de control. Una porción de control de una aplicación puede ejecutarse a alta prioridad mientras una menos importante de comunicación o almacenamiento de datos puede ejecutarse a una prioridad menor. Esto significa que múltiples ciclos PID se ejecutan a tiempo real para asegurar la estabilidad del proceso mientras una interfaz con el usuario (HIM) se ejecuta a una prioridad menor. El sistema operativo a tiempo Real garantiza que los cálculos PID obtengan recursos del procesador cuando se necesiten mientras el HMI utiliza los ciclos libres.

3.8.2 LabVIEW Distribuido para Fácil Adquisición de Datos

Compact FieldPoint incluye capacidades Ethernet, de manera que se puede distribuir fácilmente inteligencia y control dentro de la fábrica o planta de producción. Sin embargo, más allá de la habilidad de distribuir medida y control, Compact FieldPoint es también una plataforma sencilla de usar para Adquisición de Datos, y ofrece un corto tiempo hasta la primera medida. Tanto la configuración del equipo como la interfaz de programación de la aplicación (API) son simples y fáciles de usar, además el equipo cuenta con acondicionamiento de señal incorporado para conexión rápida y directa de los sensores. Por ejemplo, el módulo cFP-TC- 120 filtra una señal de termopar y la mide con un convertidor A/D Deltasigma con 16-bits de resolución. Éste lee el valor de compensación del junta fría del bloque conector, realiza los cálculos de linearización del termopar y retorna un valor de temperatura en unidades de ingeniería.

La configuración del entorno Compact FieldPoint utiliza menús en el Explorador de FieldPoint (FieldPoint Explorer). Con el explorador de FieldPoint, se configura el sistema completo, incluyendo parámetros de red, parámetros de módulos y E/S y objetos de canal con nombre.

Así mismo se pueden configurar fácilmente parámetros E/S como rangos de entrada, estados de salida iniciales de arranque, estados del temporizador de seguridad o watchdog al utilizar diálogos de ventana intuitivos. Adicionalmente, se puede probar interactivamente los módulos E/S y sus canales (usando la ventana de pruebas), observando valores de los datos de entrada y fijando los valores de salida. Finalmente el API o interfaz de LabVIEW es una arquitectura simple de lectura / escritura de un solo punto.

En LabVIEW se abre una sesión hacia la E/S de Compact FieldPoint utilizando el nombre con el que se configuró en el explorador de FieldPoint. Por ejemplo, se pueden configurar estaciones de registro de datos para monitorizar señales como temperatura, presión y flujo; rastrear la condición de válvulas; y almacenar los datos localmente o transmitirlos a través de la red Ethernet.

3.8.3 Control Embebido

Tanto el diseño mecánico, arquitectura de software como las capacidades de control de LabVIEW se combinan para hacer de Compact FieldPoint una plataforma ideal para **control embebido en tiempo-real**.

3.8.4 Arquitectura de Software

La teoría estándar de control está basada en toma de decisiones punto a punto. Debido a que Compact FieldPoint utiliza una arquitectura simple de lectura/escritura punto a punto tiene una disposición natural para control embebido. Utilizando el procesamiento punto a punto disponible en LabVIEW Real-Time se realiza procesamiento y control avanzado de señales. En este contexto el programa lee un punto a la vez y pasa cada punto dentro de funciones especiales diseñadas para realizar funciones continuas como

promedios, cálculos matemáticos, desviación estándar y cálculos de frecuencia. Mediante esta metodología se configuran fácilmente ciclos de control en tiempo real sin la complicación adicional de recopilar búferes de datos.

3.8.5 Capacidades de Control de LabVIEW

Cuando la aplicación en curso requiere potente funcionalidad y desarrollo rápido, la clave es un software flexible que se integre perfectamente con el equipo. LabVIEW es una plataforma de desarrollo gráfico estándar en la industria que proporciona todas las herramientas necesarias para crear las aplicaciones avanzadas con características completas de medida y control. LabVIEW facilita la construcción simple de aplicaciones complejas utilizando una paleta extensiva de funciones y herramientas – desde ciclos simples control PID analógico para procesos hasta **sistemas de control híbridos** con muchos canales que combinan componentes analógicos y discretos. El conjunto de herramientas de Control PID para LabVIEW contiene bloques para PID básico y avanzado, prealimentación (feedforward), y control difuso. También se utiliza para pruebas, modelación y simulación lineal y no lineal.

Cuando se planifica un sistema de control, se aprovechan las capacidades de LabVIEW para implementar controles básicos o incorporar fácilmente técnicas de control avanzado como desacoplamiento de perturbaciones, planificación de ganancias y control de lógica difusa. Para usuarios sin experiencia, hay herramientas potentes como la autosintonización PID que facilitan la iniciación al proceso de diseño.

La instrumentación virtual simplifica el desarrollo de sistemas distribuidos

3.8 Links

En este apartado se adjuntan una serie de links donde se puede acceder a multitud de información sobre sistemas embebidos a nivel software y hardware.

- Link a la página académica, que es muy interesante para conseguir información a nivel educativo: <http://www.ni.com/academic/esa/>
- En los siguientes links se encuentra abundante información relacionada con sistemas embebidos:
 - <http://www.ni.com/labview/design/esa/>
 - <http://www.ni.com/fpga/rio.htm> (Tecnología RIO)
 - <http://sine.ni.com/nips/cds/view/p/lang/es/nid/202228> (Bundles de sistemas embebidos)

Dentro de este link se accede a información más específica sobre las distintas plataformas software para el desarrollo de sistemas embebidos. Dichas plataformas corresponden a: LabVIEW Real Time, LabVIEW FPGA, LabVIEW Embedded, LabVIEW Embedded para ADI Blackfin, LabVIEW DSP y LabVIEW para Touch Panels.

- En el siguiente link se accede a la página exclusiva para desarrollos embebidos: <http://www.ni.com/labview/embedded/>

- A nivel hardware, dependiendo del software que vayamos a utilizar, podremos utilizar una plataforma u otra. Los diferentes conjuntos de plataformas hardware y software:
 - Compact RIO: Plataforma hardware que se usará con LabVIEW Real Time y LabVIEW FPGA <http://www.ni.com/compactrio/>
 - PXI ó PCI Reconfigurables: Plataforma para trabajar bajo PC de sobremesa o PXI con una tarjeta que lleva integrada una FPGA para poder descargar el código en la FPGA. Dicha plataforma podría estar trabajando bajo un sistema operativo en Tiempo Real o bajo Windows.
(<http://sine.ni.com/nips/cds/view/p/lang/es/nid/11829>)
- DSPs: La plataforma hardware se limita a los DSPs Speedy-33 y Texas Instruments DSK.
(http://search.ni.com/nisearch/nisearchservlet?q=dsp&filter=%2Btaxonomy%3Atop+%2B%28meta.collection%3Awww+meta.collection%3Awww2+meta.collection%3ApsEn+meta.collection%3Acommunity+meta.collection%3Acms%29+%2B%28language%3Aen+nilanguage%3Aen+nilanguage%3Aenglish%29+-docstatus%3Aobsolete&category=%2Btaxonomy:%22Tutorials%20and%20Application%20Notes%22&nistype=default&cluster=false&show_de=true&sortby=default&sortorder=ascending&ddown=2&view=basicsearch&nresults=10&lang=en)
- Embedded: El módulo de LabVIEW Embedded sirve para cualquier microprocesador de 32 bits, por lo que a nivel hardware podrías usar cualquier micro.
<http://community.ni.com/embedded>
http://www.ni.com/labview/embedded_dev_module.htm

4. ASPECTOS FINALES

En esta apartado se describen aspectos complementarios a la plataforma descrita que por ser un entorno abierto y flexible admite interactuar con otros entorno y herramientas que complementan y potencian su funcionalidad y operatividad sobre todo en el ámbito académico. Entre las más notables cabe referenciar:

4.1 Herramientas complementarias

El entorno visualSTATE es una plataforma de diseño gráfico basada en máquinas de estado que genera código C y C++ operativo con independencia del sistema RTO empleado. La plataforma no es totalmente UML (Unified Modeling Language) sino que está fundamentalmente orientada al control lógico mediante algoritmos expresados como máquinas de estados.
<http://www.iar.com>.

Para aplicaciones embebidas que requieran multiproceso basado es especialmente adecuado el microcontrolador de Parallax P8X32A-D40 <http://www.parallax.com/> cuyas características se adjuntan:

Package Type	40-pin DIP
Model Number	P8X32A-D40
Processors (cogs)	Eight
Architecture	32-bits
System Clock Speed	DC to 80 MHz
Power Requirements	3.3 volts DC
Clock Modes	(a) External crystal 4 -8 MHz (16 x PLL) (b) Internal oscillator ~12 MHz or ~20 kHz (c) Direct drive
Global RAM/ROM	64 K bytes; 32 K RAM / 32 K ROM
Cog RAM	512 x 32 bits each
I/O Pins	32 (simultaneously addressable by all eight cogs)
Current Source/Sink per I/O	50 mA

4.3 Entorno NI ELVIS

NI Elvis es el acrónimo de NI Educational Laboratory Virtual Instrumentation Suite. Proporciona un laboratorio completo donde realizar prototipaje electrónico, análisis test y medidas con instrumentos virtuales. Entre los numerosos links sobre esta plataforma: http://www.ni.com/academic/ni_elvis/.

4.3.1 Descripción

NI ELVIS es un entorno de prototipaje y diseño basado en LabVIEW, pensado para laboratorios de universidades de ciencias e ingeniería. NI ELVIS consta de instrumentos virtuales basados en LabVIEW, un dispositivo de adquisición de datos multifunción (DAQ) y una estación de trabajo para banco de pruebas con una tarjeta de conexiones para prototipos (protoboard). Esta combinación proporciona el conjunto completo de instrumentos, listos para usar, que se puede encontrar en cualquier laboratorio docente. Como está basado en LabVIEW proporciona capacidades completas de adquisición de datos y prototipaje, el sistema es ideal para la realización de prácticas académicas adecuadas para un amplio rango de currículums docentes. Eso no quita que NI ELVIS, gracias a su gran robustez, pueda servir también como un banco de pruebas multifuncional para uso en laboratorios industriales, donde se realicen pruebas diversas, ya sean de test o de calidad y respuesta del prototipo.

4.3.2 Características principales

Entre las características operativas y funcionales más importantes cabe destacar:

- Funcionalidad multi instrumento integrada

- Plataforma abierta basada en el software estándar de la industria LabVIEW y dispositivos DAQ de NI
- Combinación de instrumentación, adquisición de datos y estación de prototipos
- Completo abanico de instrumentos virtuales
 - Osciloscopio, DMM, Generador de Funciones, Fuente de Alimentación Variable, Analizador Bode, Generador de Formas de Onda Arbitraria, DSA, Analizador Tensión/Corriente
 - Proporcionado el código fuente de LabVIEW
 - Capacidad para particularizar en entorno LabVIEW
- Almacenamiento de datos en Excel o HTML

4.3.3 Arquitectura de NI ELVIS

NI ELVIS utiliza un software basado en LabVIEW, una tarjeta de adquisición multifunción, y una estación de trabajo personalizada para proporcionar funcionalidad a un amplio conjunto de instrumentos. La tarjeta DAQ y la estación de trabajo se comunican mediante líneas I/O digitales y un módulo de comunicaciones personalizadas basado en LabVIEW.

4.3.4 Áreas de aplicación

La combinación de la estación de trabajo, el software LabVIEW y la tarjeta de adquisición de NI ELVIS tiene una amplia aplicación en las disciplinas académicas de ingeniería. El sistema es abierto, no sólo en términos del software, sino también en su hardware de acondicionamiento de señal. Esta flexibilidad proporciona a los profesores la capacidad de implementar el sistema en varios niveles, desde el principio hasta los niveles más avanzados. Además, la naturaleza abierta del sistema permite que el profesorado pueda ofrecer de una forma sin precedentes su experiencia de primera mano a los estudiantes en sus diversas fases de formación.

4.3.5 NI ELVIS en Ingeniería

El sistema NI ELVIS se ajusta adecuadamente para la enseñanza universitaria, a distintos niveles, de la electrónica y diseño de circuitos en el ámbito de la ingeniería electrónica. La operatividad del sistema ofrece capacidades completas para verificación, medida y registro de datos; necesarios para este tipo de formación. La tarjeta de prototipos es separable y ofrece a los estudiantes la capacidad de construir circuitos en casa y emplear el tiempo de laboratorio más eficientemente.

Los instrumentos de NI ELVIS, como el analizador Bode y el trazador de curvas, ofrecen la oportunidad de enseñar cursos de nivel avanzado de análisis de señal y procesado. Los estudiantes pueden construir filtros software en LabVIEW y filtros hardware en la tarjeta de prototipos, y entonces comparar su comportamiento. Por primera vez, los estudiantes tienen la posibilidad de ver los efectos de los filtros "antialiasing" hardware sobre las señales. La tarjeta de

prototipos ofrece una plataforma ideal para construir acondicionadores de señal personalizados para varios sensores y transductores.

4.3.6 Diseño embebido y plataformas de prototipaje

En este entorno es muy adecuado para comenzar con el diseño e implementación de sistemas embebidos desde la perspectiva docente, permitiendo el:

- Diseño y análisis de sistemas empotrados
- Diseño de control
- Diseño de filtros digitales
- Diseño de circuitos electrónicos
- Diseño mecánico
- Diseño de algoritmos

En este contexto y desde la perspectiva de sistema abierto se puede enlazar con:

- **Especificaciones técnicas de NI ELVIS y tarjetas de adquisición**
- **LabVIEW.**
- **MultiCAP.** Es un potente e intuitivo programa de captura esquemática. Incluye numerosas innovaciones y características que permiten ahorrar tiempo, incluyendo trabajo sin modelado, realización automática de pistas y una amplia base de datos organizada en apartados con una estructura lógica dentro del escritorio, de esta forma, se pueden crear complejos diseños en el menor tiempo posible
- **MultiMCU.** Es un unidad de co-simulación para microcontroladores que puede ser añadida a la herramienta de simulación electrónica MultiSIM 9, en su versión educacional. Permite incluir un microcontrolador, programar en lenguaje C y ensamblador y modelar el circuito. Programar microcontroladores en el entorno de MultiSim hace que la programación sea lógica, interactiva y el aprendizaje se realice de forma más sencilla, utilizando todas las ventajas y herramientas que ofrece MultiSim9.
- **Multisim9.** Es una potente herramienta de simulación que incluye una versión completa y totalmente integrada de Multicap para la captura de esquemáticos. Verifica circuitos y localiza errores antes de que éstos se propaguen en el diseño y den lugar a problemas costosos. Investiga circuitos utilizando los instrumentos virtuales únicos de la industria y simulaciones iterativas con cambios durante la ejecución. Obtenga un mayor conocimiento del rendimiento de su circuito gracias al extenso conjunto de tipos de análisis. Multisim le ayudará a optimizar los diseños y ahorrar tiempo en su comercialización.

Multisim 9 también ofrece integración con los productos LabVIEW y SignalExpress de National Instruments, lo que permite integrar todo el

proceso de diseño y análisis. Entre las nuevas características destacables en esta versión encontramos:

- Integración con National Instruments LabVIEW y SignalExpress
 - Capacidad de crear sus propios instrumentos. Los instrumentos virtuales de LabVIEW ahora pueden utilizarse desde Multisim.
 - Intercambio de ficheros con NI LabVIEW – Multisim puede leer y escribir ficheros NI LVM y TDM:
 - Se puede exportar resultados de simulación a LabVIEW y comparar fácilmente los resultados predichos y reales.
 - Simulación de instrumentos reales de Tektronix.
 - Simulación de instrumentos reales de Agilent.
 - En el proceso de análisis se pueden introducir expresiones para particularizar los resultados a las necesidades del usuario (postprocesado).
 - Soporte para variantes de diseño.
 - Perfiles de simulación (guardar y reutilizar conjuntos de parámetros SPICE)
 - Nuevos asistentes de circuitos (Circuit Wizards) y generadores de modelos (Model Makers)
 - Potentes herramientas para el análisis del peor caso (Worst-Case Analysis)
 - Captura de esquemáticos mejorada significativamente, incluyendo soporte a buses industriales.
 - Completo soporte para diseños jerárquicos (Diseño Top-Down).
 - Amplias capacidades de anotación de circuitos.
 - Verificación de errores mejorada (soporte para pines no conectados).
 - Asistente de circuitos adicionales. Esta opción permite crear, por ejemplo, fácilmente circuitería de amplificadores operacionales y amplificadores MOSFE.
 - La fuente lineal definida por tramos soporta ficheros de datos grandes y permite hacer "bucles" de datos
 - Buen rendimiento cuando se capturan diseños grandes
 - Impresión mejorada de proyectos multipágina
- **MultiVHDL.** Es una potente herramienta usada por usuarios experimentados, pero con una excepcional facilidad de uso. MultiVHDL incluye un editor de máquina de estados, potente simulación, controlador de proyectos, un diseñador gráfico de prototipos y muchas características de depuración. MultiVHDL puede ser usado como una aplicación independiente o junto con el simulador MultiSIM usando la correspondiente co-simulación.

MultiVHDL cumple con los estándares IEEE. La co-simulación permite introducir componentes modelados con VHDL dentro de circuitos en modo mixto y comprobar el comportamiento de la FPGA/CPLD en un circuito real. Esto es necesario porque diversos componentes son modelados mejor en diversos lenguajes de simulación. En este contexto el módulo MultiVHDL y la co-simulación permiten analizar diseños combinando el modo mixto SPICE y el modelado de componentes VHDL. Esta tecnología habilita a los diseñadores a simular tarjetas completas que contienen componentes discretos, analógicos, lógicos, de lógica programable y dispositivos digitales complejos.

Entre las características más notables se destaca:

- Editor de código fuente.
 - Simulación precisa y rápida.
 - Editor de máquina de estados.
 - Diseñador de banco de pruebas gráfico.
 - Detección de errores durante la ejecución - detecta muchos errores previos a la compilación.
 - Visor gráfico de formas de onda.
 - Soporte completo a VHDL IEEE 1076-93 y 1164.
 - Depuración interactiva y avanzada a nivel fuente.
 - Co-simulación con SPICE.
- **MultiBOARD.** Es una solución avanzada y de fácil manejo para el desarrollo de placas de circuito impreso o PCB
 - **MultiROUTE.** Es una herramienta de autoruter automático. Dada la alta complejidad de los diseños PCB, el posicionamiento de componentes y pistas de forma manual termina siendo una técnica impráctica. En este tipo de diseños es donde MultiROUTE se convierte en una potente herramienta de diseño. Soporta diseños altamente complejos incluyendo tarjetas con múltiples capas y una densa cantidad de componentes.
 - **Foro de discusión y consulta de NI ELVIS.** En la base del conocimiento tecnológico de herramientas para la ingeniería electrónica. En este contexto de instrumentación virtual sobre el modelado e implementación de sistemas sobre plataformas reconfigurables los aspectos más relevantes a tratar son:
 - Analizar un problema de desarrollo de software empotrado y expresarlo de manera precisa y concisa.
 - Diseñar la estructura de un módulo software que solucione el problema, y evaluar las diferentes alternativas.

- Implementar un módulo de forma que aproveche de manera óptima los recursos.
- Trabajar en equipo, colaborando en todos los aspectos del desarrollo tanto del software como hardware.
- Aprender los aspectos de ingeniería en el desarrollo del software y hardware, como el trabajo con recursos limitados, mantenibilidad, dependencia o división del trabajo.
- Análisis y diseño. Técnicas de desacoplamiento. Dependencias. Especificación de procedimientos.
- Excepciones y precondiciones. Tipos abstractos de datos. Invariantes de representación. Iteradores.
- Modelos de objetos e invariantes. División modelo-vista-controlador. Análisis dinámico y pruebas.
- Patrones de diseño. Patrones de construcción. Patrones de comportamiento. Patrones de estructura.
- Proceso de desarrollo. Metodologías ligeras. Sistemas de control de versiones. Compilación, depuración y prueba para sistemas empujados. Gestión de errores.
- Restricciones de recursos. Optimización a nivel de la especificación. Análisis de planificabilidad y planificación de tareas.
- Sistemas operativos para sistemas empujados. Optimización a nivel del compilador. Parametrización del optimizador del compilador. Soportes de ejecución alternativos.
- Optimización del consumo de energía.