



Universidad  
Politécnica  
de Cartagena



**industriales**  
etsii UPCT

# **Programación flexible de controladores de automatización compactos comerciales para la monitorización de aerogeneradores residenciales**

---

**Titulación:** Máster en Energías Renovables

**Alumno/a:** Patricia García Conesa

**Director/a/s:** Ángel Molina García  
Manuel Jiménez Buendía

Cartagena, Septiembre de 2013

# Índice de Contenidos

---

1. Introducción y objetivos.....	4
2. Introducción a las energías renovables y sistemas de adquisición de datos .....	6
3. Instrumentación y Montaje .....	9
3.1. Configuración de la conexión del controlador CompactRIO.....	12
3.2. Programación del controlador CompactRIO .....	15
4. Metodología .....	17
4.1 Programación archivos Virtual Instruments de tarjeta de adquisición de datos .....	17
4.2 Programación archivos Virtual Instruments del controlador CompactRIO .....	22
5. Resultados .....	32
6. Conclusiones y trabajos futuros .....	36
7. Bibliografía .....	37

# Índice de figuras

---

<i>Figura 1: Esquema gráfico instalación</i> .....	4
<i>Figura 2: Representación funcionamiento FIFO</i> .....	4
<i>Figura 3: Modelos aerogeneradores de eje vertical</i> .....	6
<i>Figura 4: Esquema funcionamiento controlador</i> .....	8
<i>Figura 5: Controlador CompactRIO NI cRIO-9014</i> .....	9
<i>Figura 6: Tarjeta de adquisición de datos modelo NI9215</i> .....	10
<i>Figura 7: Software de programación grafica LabVIEW</i> .....	10
<i>Figura 8: Fuente de tensión y generador de funciones</i> .....	10
<i>Figura 9: Esquema de la instalación</i> .....	11
<i>Figura 10: Measurement &amp; Automation Explorer (MAX)</i> .....	12
<i>Figura 11: Protocolo de internet</i> .....	13
<i>Figura 12: Measurement &amp; Automation Explorer</i> .....	13
<i>Figura 13: Configuración CompactRIO con MAX</i> .....	14
<i>Figura 14: Conexión CompactRIO</i> .....	15
<i>Figura 15: Propiedades tipo de datos FXP</i> .....	16
<i>Figura 16: Configuración FIFO</i> .....	18
<i>Figura 17: Diagrama de Bloques tarjeta.VI</i> .....	19
<i>Figura 18: Panel de control tarjeta.VI</i> .....	20
<i>Figura 19: Compilación tarjeta.VI</i> .....	21
<i>Figura 20: Llamada a la tarjeta.VI</i> .....	21
<i>Figura 21: Programación iteraciones bucle sin FIFO</i> .....	22
<i>Figura 22: Programación bucle sin FIFO</i> .....	23
<i>Figura 23: Corrección fecha y hora del servidor SNTP</i> .....	24
<i>Figura 24: Sincronización servidor SNTP</i> .....	25
<i>Figura 25: Selección archivo ni-rt.ini</i> .....	25
<i>Figura 26: Corrección fecha y hora del servidor SNTP</i> .....	26
<i>Figura 27: Selección tiempo de medición</i> .....	27
<i>Figura 28: Frecuencia y periodo de muestreo</i> .....	27
<i>Figura 29: SubVI Extrae de FIFO</i> .....	28
<i>Figura 30: Programación Diagrama de Bloques Bucle con FIFO</i> .....	30
<i>Figura 31: Programación Panel de control Bucle con FIFO</i> .....	31
<i>Figura 32: Acceso a los archivos de toma de datos</i> .....	32
<i>Figura 33: Resultados medición valores sin FIFO</i> .....	33
<i>Figura 34: Resultados medición valores con FIFO</i> .....	35

## 1. Introducción y objetivos

Para mejorar las prestaciones de los aerogeneradores de eje vertical de uso doméstico, se necesita un estudio de sus prestaciones, diseño y sobre todo y lo más importante, es el rendimiento, en términos de generación de potencia. Este trabajo se va a centrar en la capacidad de generación y más concretamente en la obtención, procesamiento y muestra de datos lo más accesible, rápido y cómodo para el usuario.

Para la toma de datos en cualquier dispositivo se necesita de herramientas que faciliten ésta tarea y que permitan la toma de datos tanto en un banco de ensayos como en emplazamientos físicos, que será objeto de aplicaciones futuras basándose en lo desarrollado en éste trabajo.

El objetivo de este Trabajo Fin de Máster es realizar una aplicación que permita medir valores de tensión, intensidad y otras variables, conectando directamente el aerogenerador a un dispositivo de toma de datos, que en este caso será un controlador compacto llamado CompactRIO, programarlo gráficamente mediante LabVIEW utilizando dispositivos de lógica programable como FPGA y de una memoria FIFO.

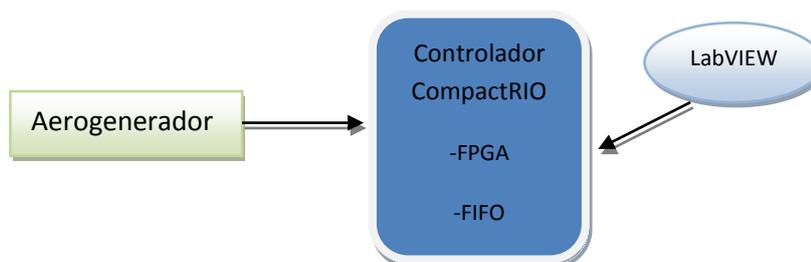


Figura 1: Esquema gráfico instalación.

El LabVIEW (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) que fue creado por National Instruments (1976), es una plataforma y un entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico de alto nivel. Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos.

Entre sus principales ventajas está la de acelerar la productividad por lo que se usa sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido. Al ser una herramienta gráfica de programación se facilita su comprensión y la creación del proyecto, es más fácil debido a la cantidad de funciones ya pre-diseñadas.

LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante National Instruments-tarjetas de adquisición de datos, PAC (controladores), Visión, instrumentos y otro Hardware- como de otros fabricantes, por lo que entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño).

Cada programa desarrollado en LabVIEW, llamados comúnmente VI debido a su extensión, consta de dos partes diferenciadas a parte del Explorador del proyecto, que engloba todo lo que hay dentro del fichero:

- **Panel Frontal:** El Panel Frontal es la interfaz con el usuario, se utiliza para interactuar con el usuario cuando el programa se está ejecutando, puedes ver datos a tiempo real. En esta interfaz se encuentra la Paleta de Controles donde se definen los controles (entradas, botones, marcadores, etc..) e indicadores (salidas, gráficas ...).
- **Diagrama de Bloques:** es un diagrama de flujo de datos que hace las funciones de código fuente. Es el programa propiamente dicho, donde se define su funcionalidad, se colocan íconos que realizan una determinada función y se interconectan. Todas las funciones se encuentran en la Paleta de Funciones.

El dispositivo a utilizar junto con LabVIEW es el controlador compacto CompactRIO, que como se verá en el transcurso de este trabajo cumplirá las expectativas de funcionalidad que se le requiere debido a la utilización de sus E/S reconfigurables mediante FPGAs y herramientas como las estructuras FIFO. Los FPGAs son chips de silicio reprogramables. Al utilizar bloques de lógica pre-construidos y recursos para ruteo programables, se utilizan para implementar funcionalidades personalizadas en hardware ya que son completamente reconfigurables y al instante toman una nueva “personalidad”. Ofrecen velocidades temporizadas por hardware y fiabilidad, pero sin requerir altos volúmenes de recursos, ya que llevan a cabo diferentes operaciones de manera paralela. Cada tarea de procesos independientes se asigna a una sección dedicada del chip, y puede ejecutarse de manera autónoma sin ser afectada por otros bloques de lógica.

Por último queda comentar ésta última estructura llamada FIFO. Sirve para transferir datos entre diferentes partes de un archivo FPGA VI, entre VIs en una tarjeta FPGA o entre dispositivos. Es una estructura de datos que aloja elementos en orden de cómo han sido recibidos y proporciona acceso a esos elementos usando la política de entrada de First-In, First-Out. Es decir, los datos que primero entran son los que primero salen.

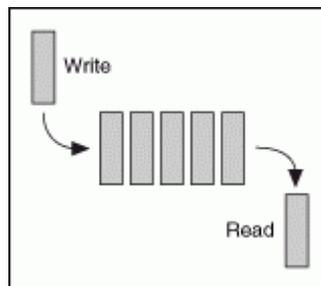


Figura 2: Representación funcionamiento FIFO.

Por lo tanto, se van a tomar datos mediante la tecnología FIFO y sin ella, a diferentes tiempos de muestreo para comparar la cantidad de muestras obtenidas. Ya que mediante FIFO nos permite tomar más datos en el mismo periodo de tiempo utilizando menos recursos.

## 2. Introducción a las energías renovables y sistemas de adquisición de datos

En los tiempos actuales de crisis energética y efecto medioambiental negativo, el uso de energías renovables está en auge. Centrándonos en la energía eólica y más concretamente en la “mini eólica” con instalaciones urbanas y de ámbito personal nos encontramos con el uso de aerogeneradores de eje vertical.

Pero este interés no es actual ya que a principio del siglo XX ya se crearon varios modelos. En primer lugar el ingeniero finlandés **Sigurd J. Savonius** en 1922 diseñó la Turbina Savonius que puede arrancar con poco viento y tiene una velocidad de giro pequeña. Su rendimiento es relativamente bajo y debido a su sencilla fabricación sus costes no son muy altos. Por otra parte años más tarde, en 1931, el ingeniero francés **Georges Darrieus** le dio nombre a su diseño. Este diseño también requiere vientos de 4 a 5 m/s, manteniendo grandes velocidades de giro y un buen rendimiento.

La unión de estos dos modelos crea un modelo que une las ventajas de ambos; el bajo autoarranque del Savonius y la eficiencia de los álabes del modelo Darrieus. Éste aerogenerador modelo DS es el que en la actualidad se están instalando en aplicaciones residenciales.

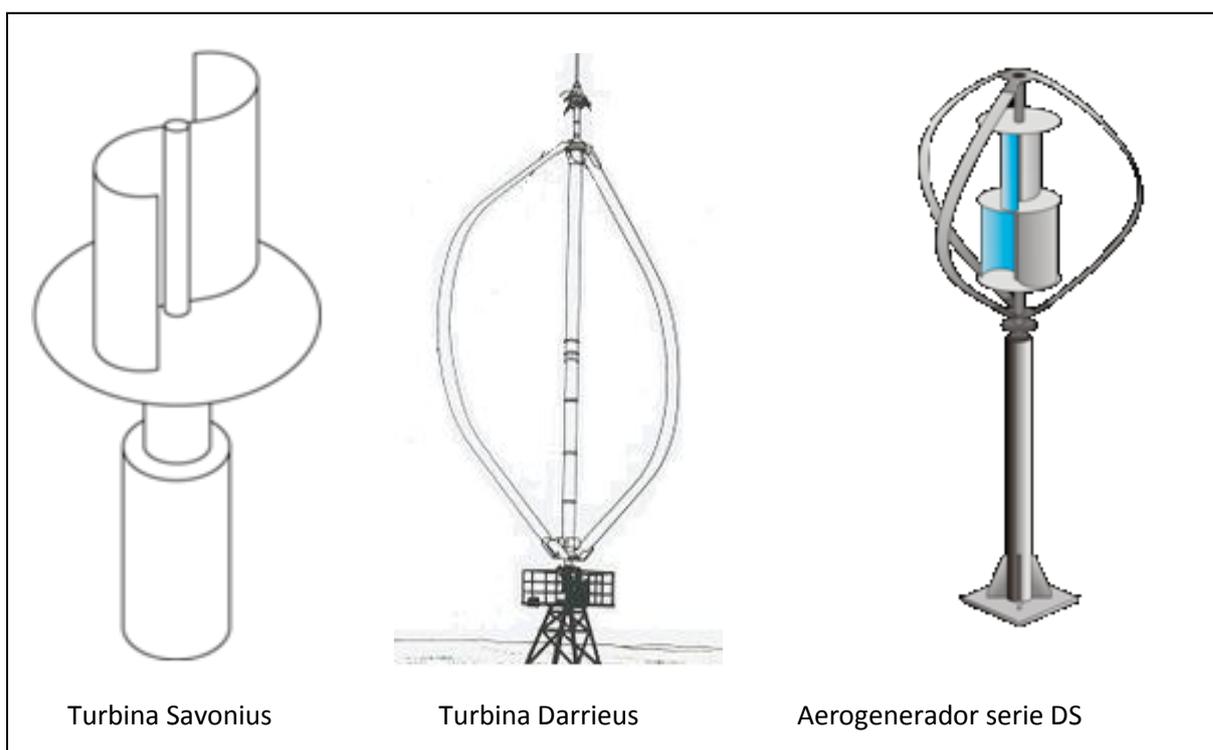


Figura 3: Modelos aerogeneradores de eje vertical.

## **Ventajas de los aerogeneradores de eje vertical**

- No requieren grandes inversiones porque sus dimensiones pueden ser reducidas. Tampoco necesitan de una torre potente o instalaciones complejas como los de eje horizontal. Su mantenimiento es sencillo, y se pueden colocar en cualquier lugar, tanto en suelo firme como en un tejado de una vivienda.
- Aprovechan mejor la dirección de los vientos, no importa si son turbulentos o de baja altura, tampoco afecta las irregularidades del terreno, que incrementan la velocidad del viento. La turbina vertical aprovecha el mínimo flujo de aire para obtener energía limpia.
- Funcionan de manera silenciosa, algo fundamental para que puedan ubicarse en lugares habitados, tanto urbanos como rurales. Tienen la ventaja de que se ponen en marcha con pequeñas velocidades del viento, a la vez que son más resistentes a las ventoleras intensas.
- Su impacto ambiental es mucho menor, ya que las aves pueden evitarlos con más facilidad y no es imprescindible que estén en espacios naturales para aprovechar la energía eólica.
- Su alineación vertical recibe al viento desde cualquier dirección, por lo que no requieren ningún sistema de alineamiento del aerogenerador, lo que reduce significativamente su coste. Esta virtud también es muy útil en lugares donde el viento cambia de forma rápida, como puede ser en medio del mar.

La era digital supuso un cambio radical en la manera en que, no hace aún muchos años, los técnicos se enfrentaban a la toma de datos provenientes de la medida de parámetros físicos. Continúan captándose "analógicamente", pero su inmediata conversión A/D ha proporcionado una mejora radical en la capacidad de procesamiento y análisis de los mismos.

Su fiabilidad, precisión, robustez, autonomía y versatilidad hacen de estos sistemas la elección más idónea para cualquier tipo de aplicación.

Existen diferentes tipos de sistemas de adquisición de datos:

- Data Loggers: Operan de manera independiente en la captura de datos en campo, que se volcarán a un ordenador posteriormente.
- Tarjetas de adquisición de datos (DAQs): Necesitan un ordenador para poder funcionar conectados mediante buses internos como PCI, PXI, PCI Express o interfaces externas como USB, RS-232 o TCP/IP.
- Controladores Compactos: Además de adquirir, generar y acondicionar la señal, con la incorporación de microprocesadores realiza análisis y control, como los autómatas programables o PLCs y los controladores de automatización programables (PAC) como CompactRIO o CompactFieldPoint de National Instruments.

Los controladores industriales de tipo compacto se llevan utilizado en diversos campos de la industria desde la década de los 60s primero con los PLCs (Programmable logic Controllers) que sustituyeron los circuitos eléctricos con relés, interruptores y otros componentes por una nueva tecnología electrónicas. Aparte de poder controlar la lógica del funcionamiento de máquinas y procesos industriales, pueden realizar operaciones aritméticas y manejar señales analógicas e incluso comunicarse con otros controladores y ordenadores de área local.

Debido al aumento de los requisitos de las aplicaciones de automatización, se necesitan controladores con una potencia de cálculo mayor y lenguajes de programación mejores y más intuitivos por lo llevo a que se desarrollaran los PAC (Programmable Automation Controller).

El PAC es un conjunto formado por un controlador, módulos de entrada y salida y diversos buses de datos interconexionados que permiten un control automatizado, diseñar prototipos y sistemas de medición. Es la unión entre la fiabilidad de control de un autómeta con la flexibilidad de monitorización y cálculo de un PC pudiendo introducirle un dispositivo electrónico FPGA (Field Programmable Gate Array) que gracias a su velocidad y la capacidad de personalización, logran que el sistema completo sea una herramienta muy potente de sistemas de adquisición de datos.

Aunque hay dos tipos de controladores PACs, los dos se programan utilizando el entorno de programación grafica de LabVIEW que reduce el tiempo de formación para su programación y facilita la creación de aplicaciones.

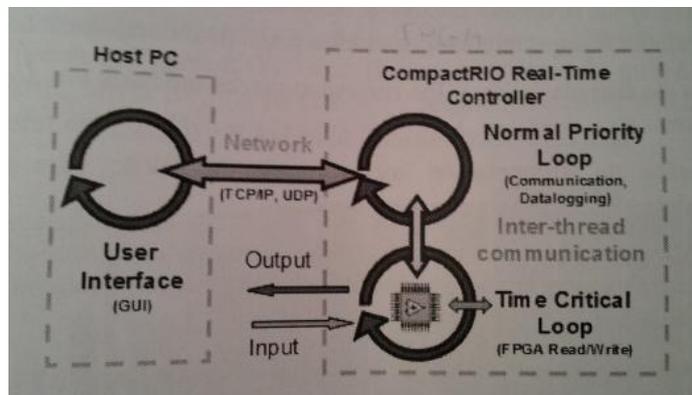


Figura 4: Esquema funcionamiento controlador.

### **3. Instrumentación y Montaje**

En el apartado anterior se ha nombrado un tipo de controlador compacto, es el CompactRIO que ha sido el elegido para utilizarlo en el montaje del proyecto debido a que es un sistema robusto de control y adquisición industrial que incluye un procesador en tiempo real y una FPGA para desarrollar aplicaciones embebidas o distribuidas, y módulos industriales de E/S intercambiables en funcionamiento que se pueden conectar directamente a sensores y actuadores.

Al combinar un procesador embebido de bajo consumo y tiempo real con una FPGA de alta prestación permite análisis de datos en tiempo real, postprocesamiento, registro de datos y comunicación con un ordenador en red de muy alta velocidad y rendimiento. Es decir, con una arquitectura de bajo coste con acceso directo a los recursos de hardware de bajo nivel, se crea un sistema de adquisición de muy altas prestaciones.

Existen dos modos de programación del CompactRIO, como ya se ha dicho que se van a utilizar los recursos de la FPGA, el modo de programación del CompactRIO será a través de FPGA Interface Mode, en donde los módulos de E/S están en el árbol de la FPGA, para hacer uso de ésta.

Los instrumentos y equipos de programación que serán necesarios para llevar a cabo el estudio son los que se nombran a continuación.

- Ordenador conectado a internet mediante un cable red.
- Controlador CompactRIO cRIO-9014.
  - o CPU interna.
  - o Chasis con 8 ranuras para insertar tarjetas de adquisición.
  - o Dos tarjetas de adquisición de datos NI 9215 con “Screw Terminal”, con cuatro canales de dos pines.
  - o Fuente de alimentación de continua de 18 W y 24 VDC.



Figura 5: Controlador CompactRIO NI cRIO-9014.

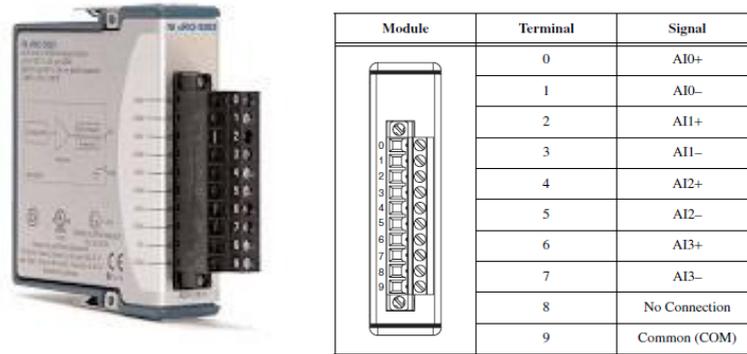


Figura 6: Tarjeta de adquisición de datos modelo NI9215.

- Software de National Instruments: LabVIEW versión 8.5.1. o posterior, los drivers del CompactRIO y de las tarjetas de adquisición de datos NI-RIO 2.4.1.



Figura 7: Software de programación grafica LabVIEW.

- Fuente de tensión, generador de funciones y cables de conexionado.



Figura 8: Fuente de tensión y generador de funciones.

Una vez nombrado y explicados los elementos que van a ser requeridos para éste montaje, el esquema de unión empieza por el ordenador.

Una vez instalado el software de National Instruments y los drivers correspondientes a la versión del software LabVIEW instalado se establece la conexión a internet mediante un cable de red. El siguiente paso corresponde al CompactRIO, se ensamblan las tarjetas de adquisición de datos cada una en su ranura correspondiente dentro del chasis, mientras se conecta la fuente de alimentación que aparece en la Figura 9 al controlador en los polos y se enchufa a la corriente. Para que haya comunicación entre el controlador y la tarjeta de adquisición de datos con el ordenador se realiza en red como ya se explicará en un apartado posterior, por lo que gracias al puerto Ethernet del CompactRIO se conectara a red también.

El último paso está relacionado con la entrada de datos, en este punto se necesitará tanto la fuente de tensión como el generador de funciones para comprobar si los valores introducidos a través las tarjetas y generados por éstos dispositivos dan el resultado positivo esperado con respecto a la programación realizada mediante el lenguaje LabVIEW.

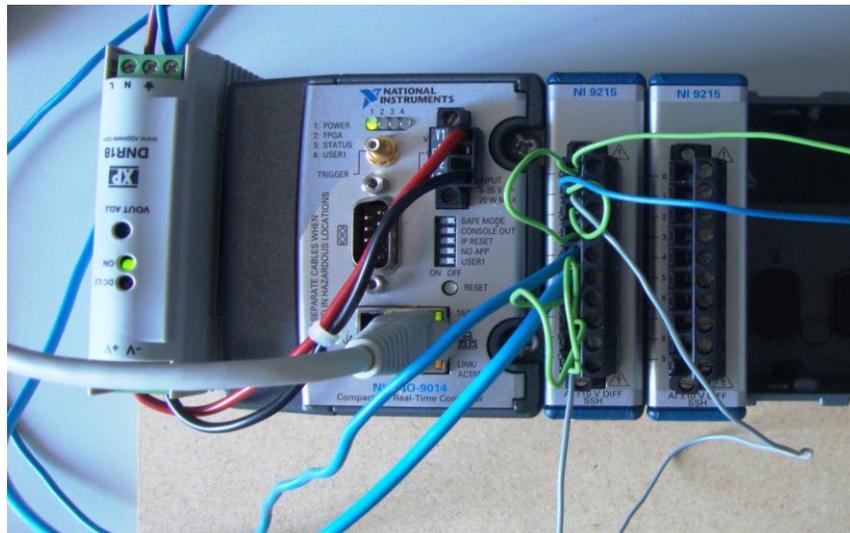


Figura 9: Montaje de la instalación.

### 3.1 Configuración de la conexión del controlador CompactRIO.

Para establecer la unión entre los dos elementos que están separados físicamente, el controlador CompactRIO y el ordenador, existen varias maneras de establecer un canal de comunicación entre ambos, conectándolos a través de una red de área local o a través de internet directamente, en nuestro caso lo conectaremos directamente a la red de área local con una IP estática.

Para lograr este propósito, aparte de conectar ambos dispositivos mediante un cable directamente a internet, se necesita la ayuda de una herramienta de National Instruments llamada Measurement & Automation Explorer (MAX), con la cual se configuran conjuntamente los dos dispositivos.

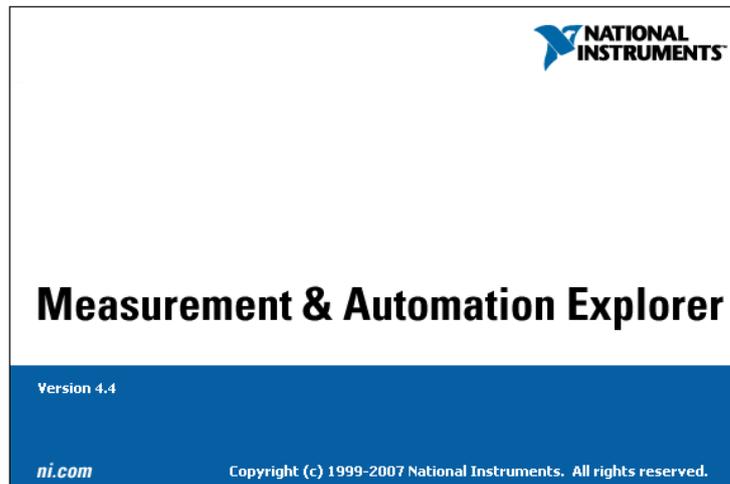


Figura 10: Measurement & Automation Explorer (MAX).

Es recomendable formatear el CompactRIO, desplazando el botón “Safe Mode” a la posición On y se presiona el botón “Reset”, ya que solo se puede hacer en ese modo. Se despliega el menú con el botón derecho sobre el icono del CompactRIO en el MAX y se selecciona “Format Disk”. Cuando termine se desactiva el modo seguro, se resetea de nuevo y se reinicia el controlador.

Desde la parte del ordenador, debemos seleccionar la dirección IP mediante la cual se va a establecer la conexión, una vez que comprobamos que está conectado a internet.

- Mediante Inicio de Windows- Configuración- Conexiones de red y botón derecho sobre propiedades, seleccionar la opción Propiedades de Protocolo internet (TCP/IP).  
Introducir la dirección IP192.168.2.250 y se obtendrá también la IP de la Máscara de Subred.

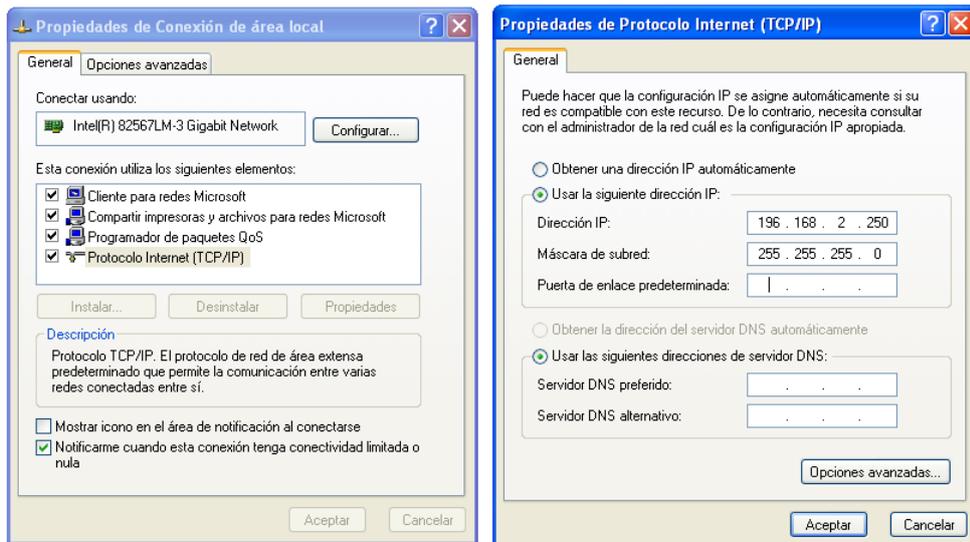


Figura 11: Protocolo de internet.

Para el controlador la establecemos a través de la herramienta MAX, situada en Tools- Measurement & Automation Explorer dentro del programa LabVIEW.

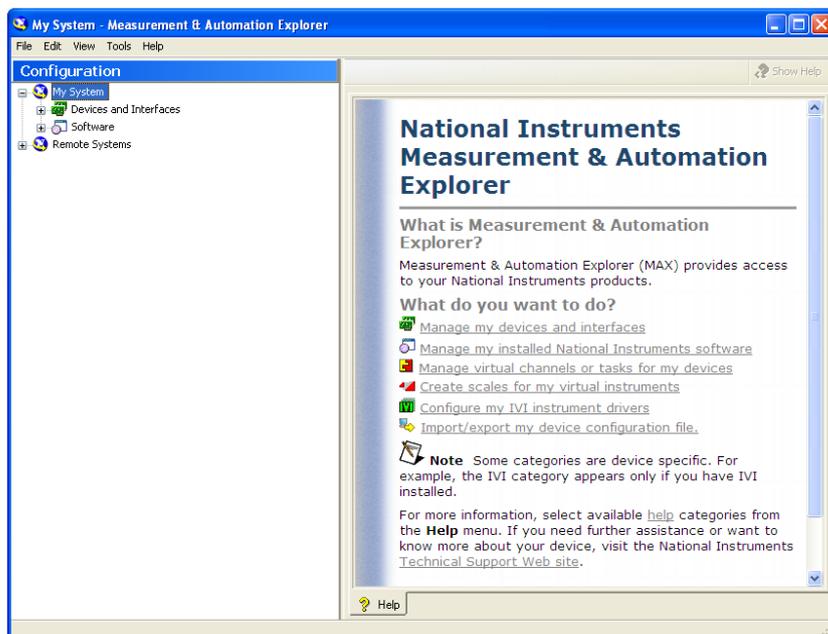


Figura 12: Measurement&Automation Explorer.

- En el apartado de Remote System es donde deberá aparecer el dispositivo Compact RIO, una vez que lo hemos conectado también directamente a internet.

Si no aparece botón derecho “Create new” y aparecerá una ventana donde se selecciona Remote Device (not on the local area) e introduciremos la IP de nuestro controlador Compact RIO que será la misma que hemos seleccionado arriba.

En la pantalla principal de MAX, dentro del apartado IP Settings, seleccionamos el botón de “Suggest Values”. En la ventana que aparece, debemos elegir la IP del host, es decir del PC, que previamente hemos designado arriba IP 192.168.2.250.

Automáticamente se establece el resto de direcciones IP para la Máscara de Subred, Gateway y DNS Server.

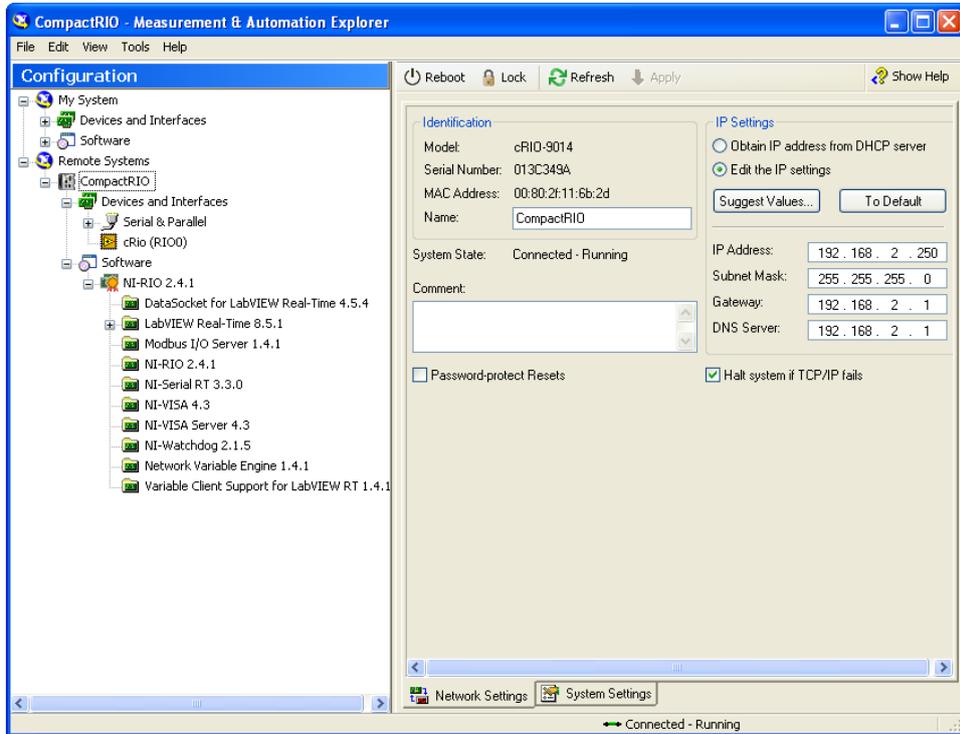


Figura 13: Configuración CompactRIO con MAX.

Seleccionar el botón Apply y esperar a que se inicialice y finalmente aparezca que está conectado y corriendo. Deberá aparecer en Devices and Interfaces, Serial & parallel y dentro de este el icono de cRIO.

- El último paso es agregar el software en sus respectivas versiones, mediante el botón derecho encima del icono de software dentro de Compact RIO y seleccionar Add/remove Software.

Como se ha dicho anteriormente deberá estar instalado la versión del LabVIEW 8.5.1 y todos los complementos y drivers necesarios en esa versión o anterior NI-RIO 2.4.1.

Por lo tanto aparecerá todo como se muestra en la Figura 13.

Completados estos pasos, ya tenemos establecida la comunicación entre los dos elementos principales del montaje.

## 3.2 Programación del controlador CompactRIO.

Antes de comenzar con la programación más importante, se requiere una configuración o programación previa del Compact RÍO, es decir que para que la unión sea efectiva en Lobbies debe ser reconocido y aparecer conectado tanto el propio controlador como los elementos que incluye y el dispositivo FPGA.

Por lo que los pasos a seguir se muestran a continuación:

- Se ejecuta el programa LabVIEW a través de su icono y se selecciona la opción “Empty Project”.
- Sobre “Project: Untitled Project 1” se hace click con el botón derecho y seleccionar “Targets and Devices”.
- Dentro de las opciones, se selecciona la opción “Real-Time CompactRIO” y luego sobre “CompactRIO”.

Una vez hecho esto, como se muestra en la Figura 14, en el árbol del proyecto aparecerá el CompactRIO bajo el modo de programación FPGA con sus componentes:

-Chasis.

-Mod1 y Mod 2. Son las dos tarjetas de adquisición de datos y dentro de cada uno habrá cuatro canales donde se conectaran físicamente mediante cables las magnitudes a medir.

-Reloj interno de frecuencia de 40 MHz.

Como se ve, englobando las partes del controlador se encuentra la tecnología FPGA, que es la que vamos a utilizar que la hemos nombrado en el apartado anterior.

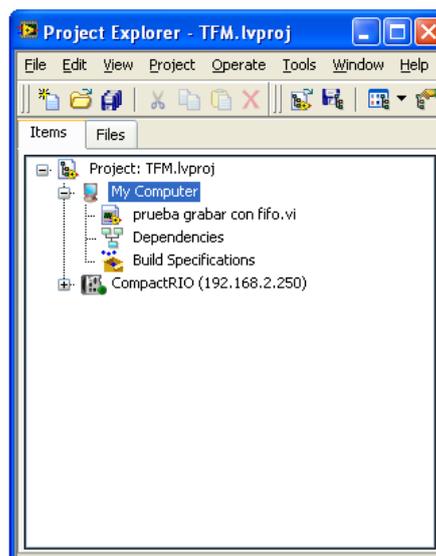


Figura 14: Configuración CompactRIO con MAX.

El último paso antes para comprobar que el controlador funciona correctamente es conectarlo haciendo botón derecho encima de CompactRIO (192.168.2.250) y seleccionar la opción *Connect*. Una vez hecho esto, aparecerá un indicador de forma circular sobre el dispositivo de color verde más claro, como indicando que esta encendido según Figura 15.

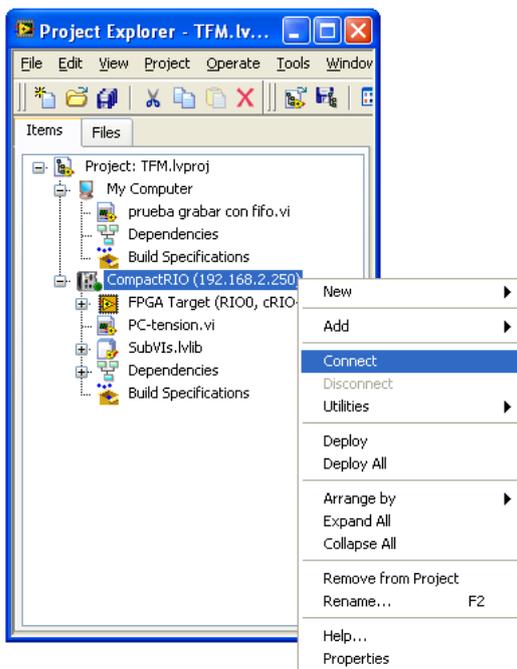


Figura 15: Conexión CompactRIO.

## 4. Metodología

### 4.1 Programación archivos Virtual Instruments de tarjeta de adquisición de datos.

En la tarjeta de adquisición de datos se programa la forma de tomar los datos; se quieren tomar datos de unas variables definidas de tensión e intensidad y otras dos a elegir para los diferentes usos de éste controlador mediante los cuatro canales a dos tiempos diferentes, es decir con una frecuencia de muestreo distinta.

Para conseguir eso, utilizaremos el propio poder del controlador y de otra manera mediante un FIFO (buffer de memoria), que es una herramienta, como ya hemos nombrado antes, que permite la adquisición de una gran cantidad de datos a una frecuencia más alta y con la posibilidad de variar la forma de procesar esos valores.

Por lo tanto se va explicar la programación que se ha realizado, con los comandos utilizados en ambas área de trabajo de LabVIEW:

-Creación archivo.VI: Dentro del árbol del proyecto con el botón derecho donde aparece FPGA target (RIO0, cRIO-9104), seleccionamos New y guardamos con el nombre que queramos, en este caso va a ser tarjeta.VI.

#### Diagrama de bloques

-Bucle *While Loop* para que se repita la acción de toma de datos hasta que se cumpla la condición de parada, que en este caso es con un control *Booleano* de Stop.

-Estructura de Secuencia *Flat Sequence Structure*. Se incluye dentro del bucle, para que lo que se programe dentro de cada *Frame* (que se consideran como subdiagramas) se ejecute según una secuencia, es decir uno tras otro.

- **Frame 1:** Un control *Count* donde se introduce elegido por nosotros, el número de ticks de reloj (4000) que se establece para temporizar el bucle uniéndolo a un *Loop Timer*, que temporiza el bucle para ejecutarlo en un intervalo de tiempo.
- **Frame 2:** Se necesita abrir los canales que van a medir y leer en continua. Desde la pantalla Project Explorer pinchamos y arrastramos la tarjeta donde está el canal o los canales que necesitemos usar (Mod1/AI0, Mod1/AI1, Mod1/AI2, Mod1/AI3). Para identificarlos mejor, se le cambia el nombre a los canales, pinchando encima desde el

Project Explorer como Tensión, intensidad y dos variables de reserva. El icono creado automáticamente se llama *FPGA I/O Node* el cual toma los valores de la tarjeta y los transmite mediante la extensión FXP (punto fijo).

A partir de aquí se empiezan dos caminos para la programación con y sin FIFO, es decir se utilizarán funciones diferentes.

### A) Toma de datos sin FIFO

Si queremos que los valores sin pasar por la FIFO salgan por pantalla conectamos directamente cuatro indicadores conectados directamente a la salida de la tarjeta. El tipo de datos es FXP, que dentro de sus propiedades asignaremos un valor de 20 bits para la longitud total del número y de 5 bits para la parte entera, por lo que mediremos valores con decimales.

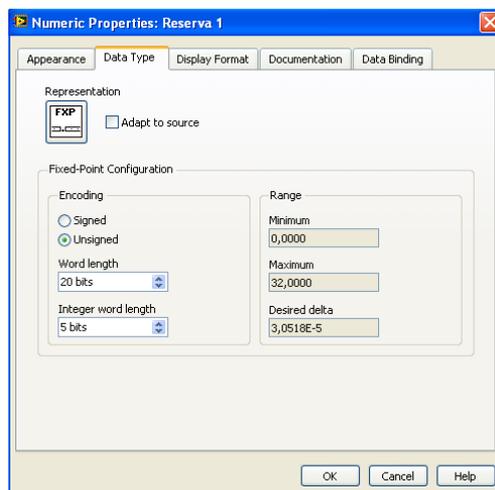


Figura 16: Propiedades tipo de datos FXP.

### B) Toma de datos mediante FIFO

Para continuar por la FIFO se necesita crear primero un *Array*, que son elementos de datos del mismo tipo desde una a muchas dimensiones y conteniendo hasta  $2^{31}-1$  elementos en cada dimensión, se utiliza la función *Build Array* uniendo cada canal de dimensión 1D se obtiene un array de 1D con los 4 canales donde se agrupan todos los valores.

Por lo tanto se necesita cambiar el tipo de dato de la tarjeta a un tipo numérico como DBL o en este caso a U32 (entero sin signo de longitud de 32 bits), utilizando *Number to Boolean Array* unido a *Boolean Array to a Number*.

Una vez creado el Array se une a un bucle *For* para fijar un número de veces fijado inicialmente lo que quiero que se repita dentro del bucle. Para utilizar el FIFO, utilizaremos los métodos llamados *FIFO Write* y *FIFO Read* pero primero hay que crearlo.

### Creación del FIFO

En Project Explorer con botón derecho sobre la FPGA-New-FIFO, aparece una ventana de propiedades donde se puede cambiar el nombre, el tipo, la capacidad de almacenamiento. Cuando configuras una FIFO, hay que especificar la implementación que determina que recurso de la FPGA tienen y la transferencia de datos por la FIFO. Se ha elegido el modo “Between the host computer and the FPGA”, Direct Memory Access (DMA) es un método con base en la FIFO de transferencia de datos entre una tarjeta FPGA y el del ordenador. DMA no implica el procesador del ordenador por lo tanto es el método disponible más rápido para transferir grandes cantidades de datos entre ambos. Con este tipo se tiene acceso para transferir datos desde los archivos Vis de la tarjeta FPGA a los archivos VIs del ordenador y viceversa.

*Type:* Target to Host-DMA

*Memory Type:* Por defecto se asigna *Block Memory* a la manera de almacenamiento de la FPGA, porque utiliza bloques de memoria embebidos.

*Depth:* La capacidad de la FIFO.

*Arbitration:* Se debe elegir *Arbitrate if Multiple Requestors only* cuando vayamos a usar cualquiera de los métodos de *Write* o *Read* junto con un bucle *Timed Loop*.

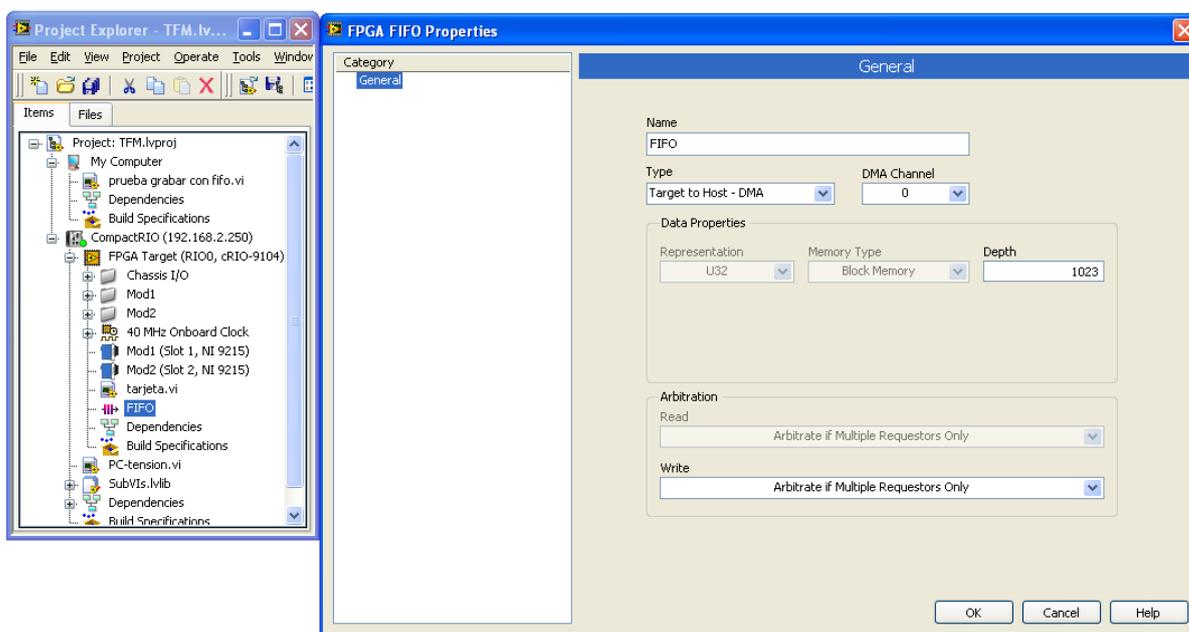


Figura 17: Implementación de la FIFO.

Desde allí lo arrastramos al diagrama de bloques y lo colocamos dentro del bucle *For* y se selecciona por defecto *FIFO Write*. Mediante esta función se van escribiendo los valores dentro del a FIFO, por lo que unimos el Array antes creado al terminal *Elements* para que vaya tomando los valores de la tarjeta y en el terminal *TimeOut* se introduce el número de ticks del reloj que la función espera cuando la FIFO está llena, para un hueco disponible en la FIFO. Se puede poner un valor por defecto como 0 para que no espere y sobrescriba los valores o un valor de -1 como es el caso, para configurar que espere indefinidamente hasta que se saquen los valores de la FIFO.

Finalmente el resultado será el que aparecer en la Figura 18.

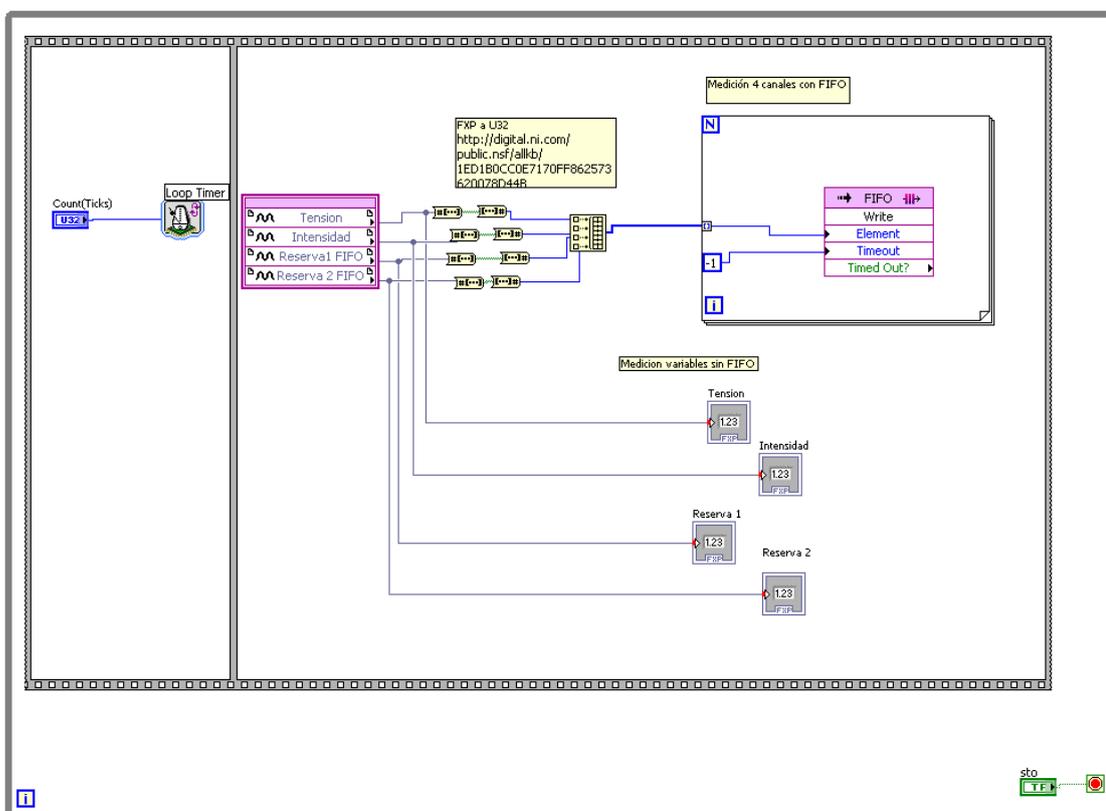


Figura 18: Diagrama de Bloques tarjeta.VI.

### **Panel de control**

De lo que se ha programado en el diagrama de bloques se representa en el panel de control según muestra la Figura 19.

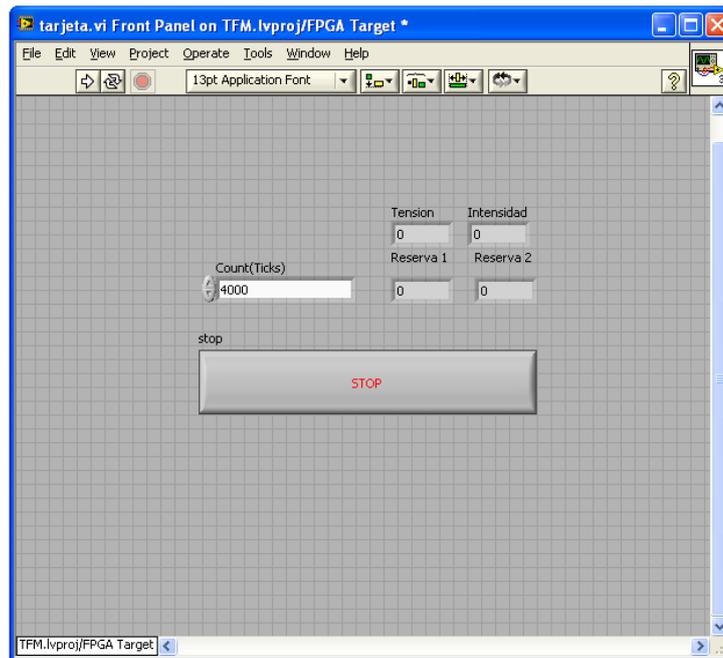


Figura 19: Panel de control tarjeta.VI.

Aparecen el control Count (Ticks), los cuatro indicadores que mostrarán en pantalla lo medido en cada canal y el botón de Stop del bucle While.

### Compilación

Una vez programado es requisito obligatorio compilar éste archivo de las tarjetas de adquisición de datos antes de empezar a medir para que pueda ser llamado por el VI del CompactRIO cuando lo programemos. En el Project Explorer con botón derecho sobre el archivo de la tarjeta y se selecciona *Compile* hasta que aparezca el mensaje que ha sido finalizado con éxito. Esta acción, se llevará a cabo de forma automática cada vez que modifiquemos algo del archivo tarjeta.VI y se quiera grabar los cambios.

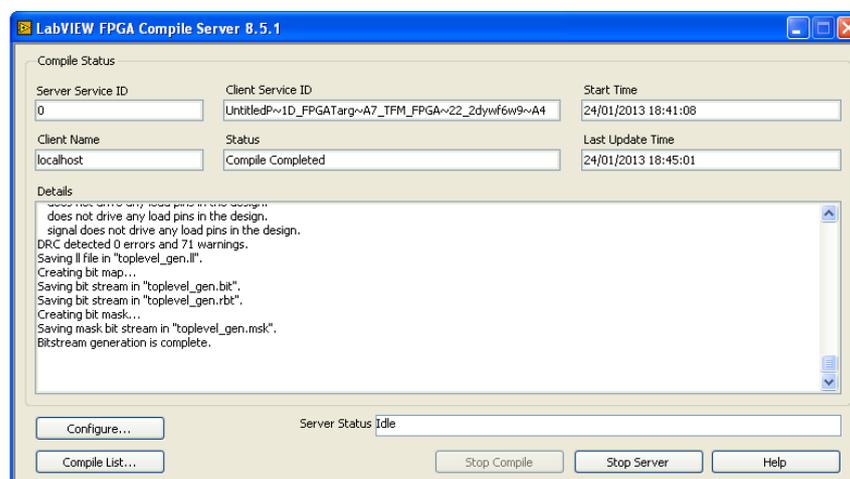


Figura 20: Compilación tarjeta.VI.

## 4.2 Programación archivos Virtual Instruments del controlador CompactRIO.

Se realiza de manera semejante a la anterior, llamando al nuevo archivo PC\_tensión.vi y creado dentro del cuerpo del CompactRIO (192.168.2.250) pero en la rama principal.

### Diagrama de Bloques

-Abrir o llamar al VI de la tarjeta de adquisición

Se va a utilizar la función *Open FPGA VI Reference (FPGA Interface)* que está en la paleta de funciones FPGA Interface, dentro de una estructura de secuencias para que sea lo primero que se ejecute dentro de éste archivo. Se colocará una variable local de un control booleano de *Stop* que se creará dentro del bucle *Timed Loop* que se creará a continuación. Ésta acción sirve para unir todos los botones de *Stop* en uno solo y que cuando se ejecute la opción de parada, al estar unidos se paren todos. Hasta que no le insertemos la tarjeta de adquisición, no habrá comunicación aún, por lo que arrastramos del explorador del proyecto FPGA Target (RIO0, c-RIO 9104).

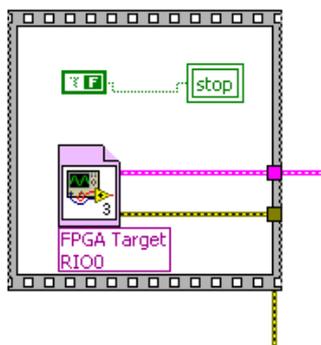


Figura 21: Llamada a la tarjeta.vi.

-Creación bucle *Timed Loop*

Se inserta esta función para ejecutar con un período especificado, cuando se va a desarrollar varios VI con diferentes regímenes. En este caso se usa en un FPGA VI, el bucle se ejecuta según el reloj interno de la FPGA.

Se va a establecer la prioridad de ejecución con un período de 1000 ms a una frecuencia de 1kHz, para que cada segundo se ejecute el bucle y compruebe que está en el segundo que se establecerá más adelante de toma de lecturas. Se cablea con la estructura de frecuencias anterior.

Dentro de este bucle es donde se programará según los criterios de tiempo y frecuencia nombrados en los objetivos con y sin FIFO.

### A. Medición sin FIFO 1 muestra cada 6 segundos.

Primero se determinará la pauta a seguir de medición, ya que nuestro objetivo es tomar una medida cada 6 segundos. Por lo tanto según muestra la Figura 22, hacemos una sencilla operación aritmética dividiendo el número de iteraciones *Loop Iteration* del bucle *Timed Loop* que se ejecuta cada segundo, entre 6 con la condición de que cuando el resto de esa división sea 0 es un instante de muestreo. Esta condición la uniremos al terminal de la estructura de casos *Case Structure* para determinar qué caso de ejecuta.

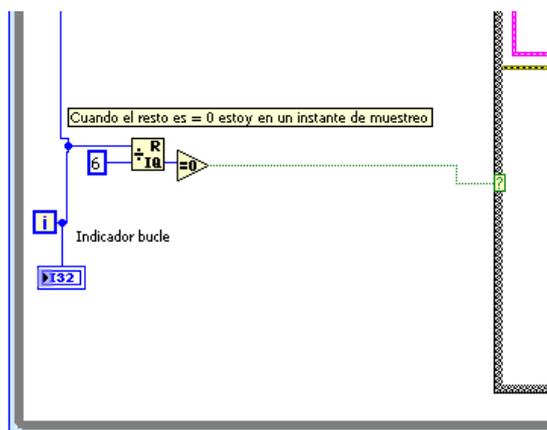


Figura 22: Programación iteraciones bucle sin FIFO.

Como lo hemos unido al caso *True*, aquí es donde se insertará la función *Read/Write Control* de la paleta de *FPGA Interface* para poder leer los cuatro canales de la tarjeta. En este caso queremos que se guarden los valores obtenidos asociado a la fecha y la hora que se toma la medida en un *.txt* en vez que aparezcan por pantalla mediante la unión a un indicador.

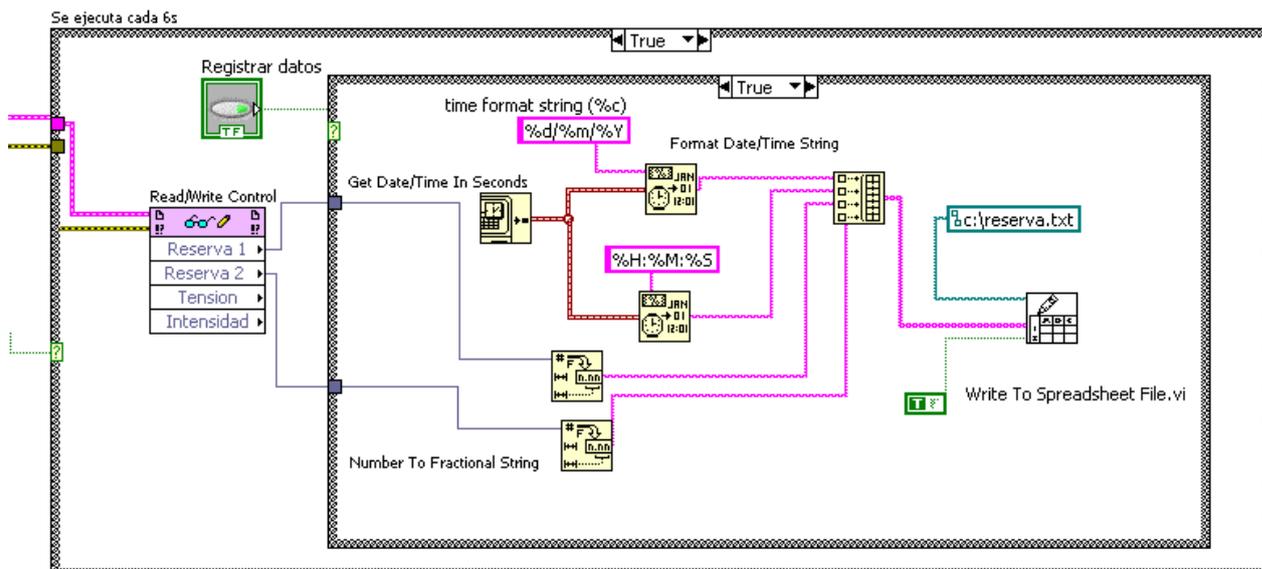


Figura 23: Programación bucle sin FIFO.

Se ha creado una estructura de casos *Case Structure* para tener la oportunidad mediante un booleano de elegir si grabar los datos o no, solo con pulsarlo.

Mediante las funciones que aparecen en la Figura 23 se muestra como crear dos Arrays, uno de fecha y otro de hora para construir el Array 1D con los valores obtenidos por ejemplo de dos canales solo. Se escriben dos *String* con el formato de fecha y hora elegido con el carácter % delante y se unen a la función *Format Date/Time String* donde se le une al terminal, la función *Get Date /Time in Seconds* que devuelve una impresión de la hora actual.

Se han convertido en *String* (cadena de caracteres) para que puedan ser imprimibles por pantalla.

Para obtener los datos correctos de fecha y hora se necesita la sincronización mediante un servidor SNTP.

### Sincronización servidor hora SNTP

Se necesita aparte de la programación en el archivo .VI, una sincronización mediante la conexión a un servidor de hora SNTP.

Buscando servidores de hora NTP aparece una lista que corresponde a España que es donde estamos y elegimos el que tiene la dirección IP 80.28.229.87.

- 1) Se accede al archivo ni-rt.ini que está alojado en el controlador. Desde MAX, haciendo click con el botón derecho sobre el CompactRIO se selecciona *File transfer*.

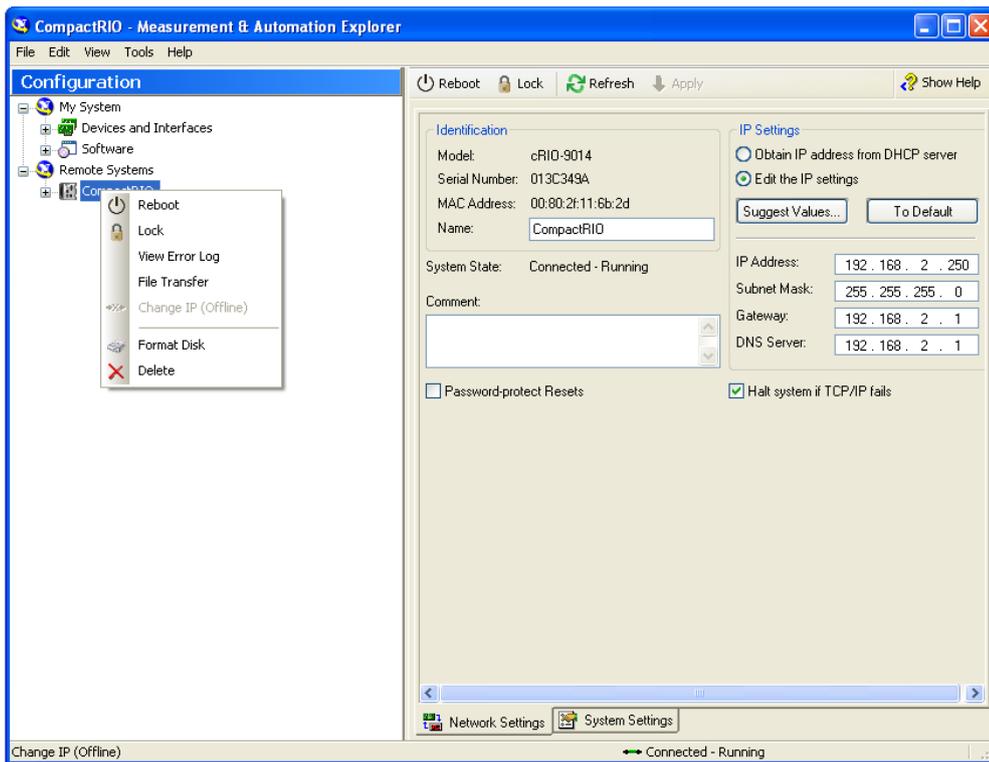


Figura 24: Sincronización servidor SNTP.

- 2) Aparece la siguiente ventana donde se selecciona el archivo ni-rt.ini y lo guardamos en el escritorio mediante el botón *To Local*.

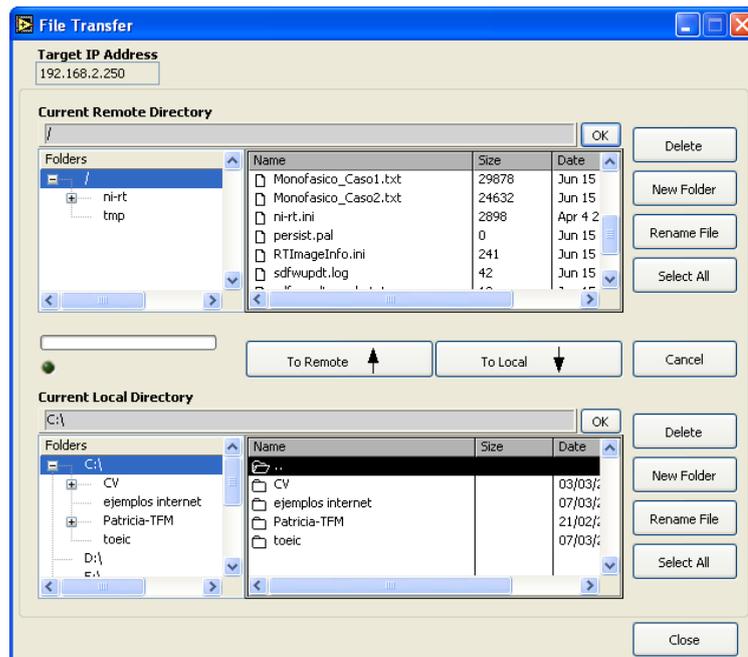


Figura 25: Selección archivo ni-rt.ini.

- 3) Se abre el archivo en el escritorio y se le añade las siguientes líneas de programación al final de la hoja de datos. Donde la dirección `x.x.x.x` será la correspondiente al servidor de hora 80.28.229.87. Una vez guardado el archivo se realiza la acción a la inversa se reescribe el archivo subiéndolo al directorio remoto (CompactRIO) mediante el botón *To Remote*.

```
[TIME SYNC]
source_priority = sntp;rtc;
source.sn timer.enable = true
source.sn timer.address = x.x.x.x
source.sn timer.interval = 60
source.sn timer.port = 123
source.sn timer.verbose = false
source.sn timer.log = false
```

- 4) Se cierra la ventana y desde la herramienta donde estamos se hace clic en *Reboot* para que el CompactRIO implemente las modificaciones hechas y automáticamente se sincronizara con el servidor SNTP.

Al comprobar la fecha y la hora en los resultados, se ha comprobado que no estaba correctamente sincronizado por lo que se le ha añadido las funciones mostradas en la Figura 26. Dentro del bucle *Timed Loop*.

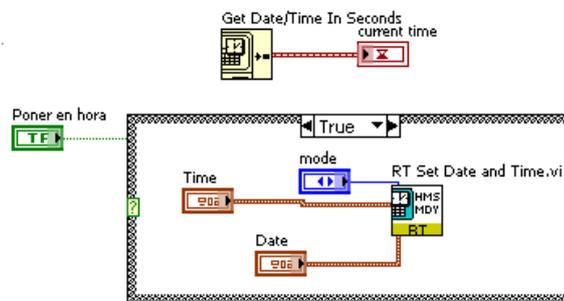


Figura 26: Corrección fecha y hora del servidor SNTP.

El último paso sería el salvado de datos mediante la función *Write to Spreadsheet File.vi* que convierte el Array 1D construido anteriormente con la función *Build Array* en una cadena de caracteres llamadas *String* y lo escribe en un archivo *.txt*. Se unen al terminal **Append to file?** un booleano de verdadero porque manda los datos a un archivo creado mediante el otro terminal **file path** que muestra la ruta del archivo guardado en la dirección IP donde se comunican ambos dispositivos.

Por lo que para ver los resultados, solo es necesario introducir la dirección IP en cualquier explorador de internet.

### B. Medición con FIFO 100 muestras a 1kHz durante 60seg.

De manera análoga a lo hecho sin la FIFO, se va usar una estructura de casos unida a la condición de iteración para fijar la condición de medida. En vez de que se ejecute cada 6 seg, que sea cada 60 seg.

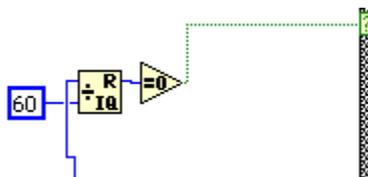


Figura 27: Selección tiempo de medición.

En este caso es necesario controlar la frecuencia de muestreo en relación con el reloj interno de la FPGA ya que se utiliza la FIFO para aumentarla. Mediante un control introduciremos el valor de 1 kHz y lo dividiremos entre 40 MHz que es la frecuencia de la FPGA obtendremos 40 kHz de frecuencia. Por lo que mediante la función *Read/Write Control* escribimos los pasos de reloj a los que queremos que se tomen las medidas, que se unirá al método de la FIFO Read.

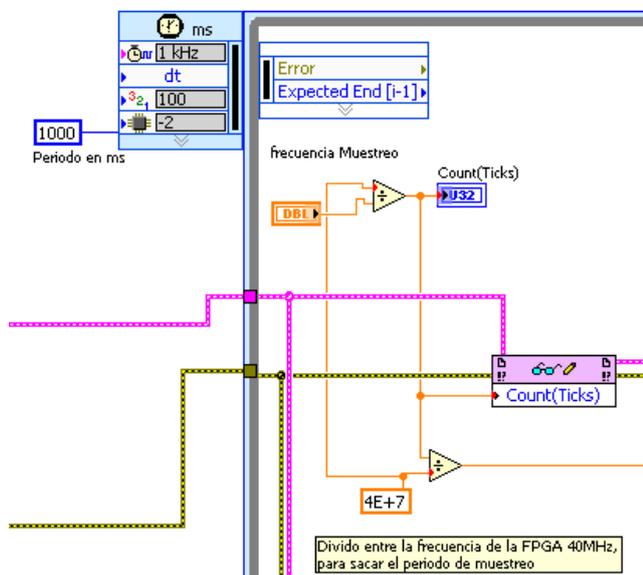


Figura 28: Frecuencia y periodo de muestreo.

Siempre que se utiliza una FIFO se necesitan dos métodos, **Write** y **Read**. El primero se implementa en el archivo de la tarjeta de adquisición y el método Read dentro del controlador solo con arrastrarlo desde el explorador del proyecto. Este método devuelve valores cuando el número de elementos está disponible (la FIFO ha llenado su memoria) o cuando el periodo de espera termina.

Se le une al terminal:

- Number of Elements: Serán 400 elementos, 100 por cada canal, ya que tomas 100 muestras a la frecuencia de muestreo de 1kHz, siendo siempre un valor múltiplo de 4.
- Timeout: Es el número de ms que el método Read espera a que estén los 400 elementos, siendo un valor de 5000 por defecto.
- Data: Devuelve los valores de los datos guardados en la memoria de la FIFO.
- Elements Remaining: Son los elementos que se quedan esperando en la memoria hasta que se lean. Al leer solo los 400 primeros se quedarán valores en la FIFO, ya que tiene una capacidad de 10000 elementos.

Por lo tanto los valores obtenidos, se conectan a un subdiagrama VI llamado Extrae de FIFO, que permitirá transformar el tipo de dato obtenido de U32 a SGL (punto flotante simple).

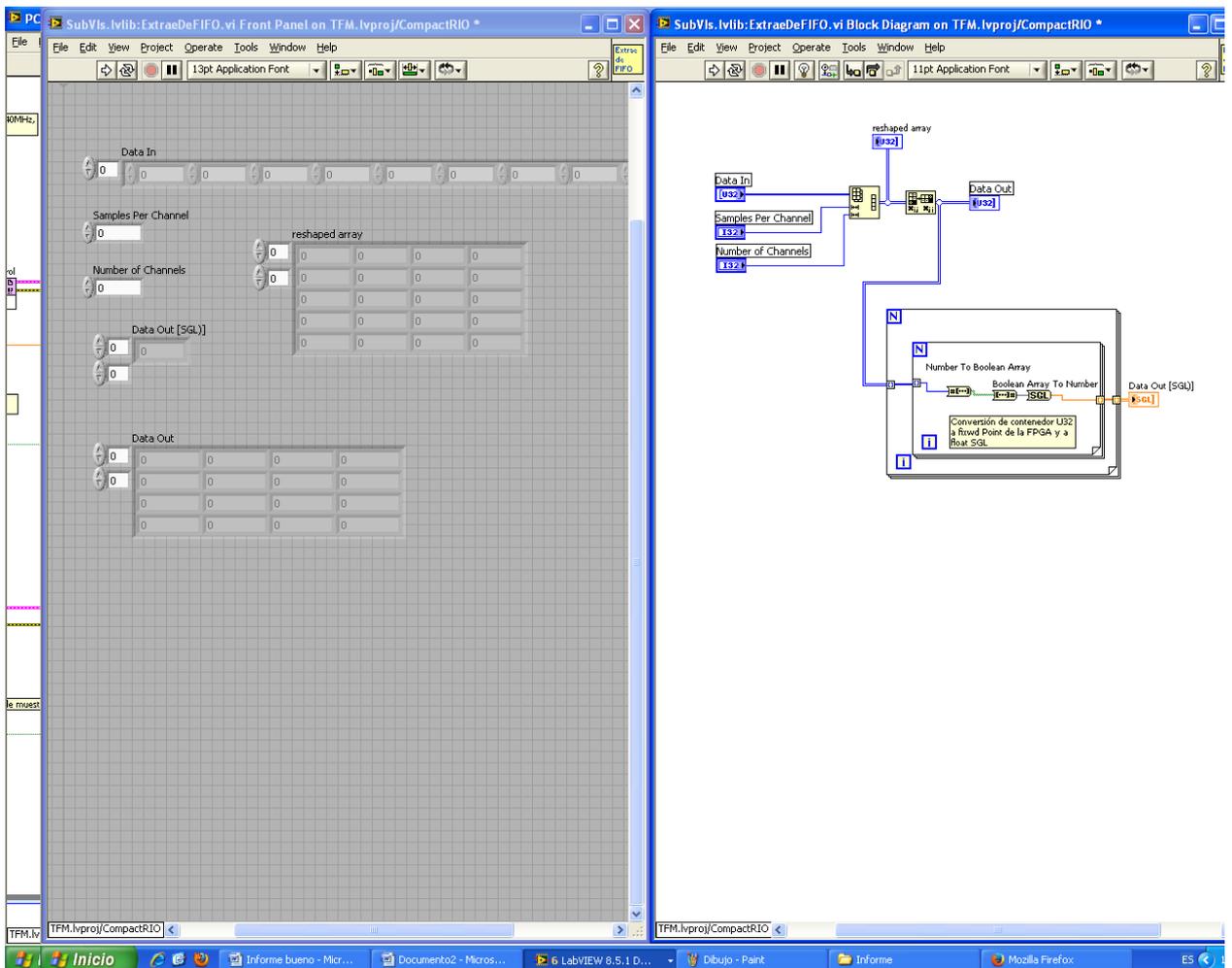


Figura 29: SubVI Extrae de FIFO.

El fin de esta transformación es la posibilidad de poder representar los valores temporalmente en una grafica. Por lo que usamos la función *Bundle* que crea un Cluster (colección ordenada de uno o más elementos que pueden ser del mismo o diferente tipo) de elementos individuales.

Se conecta primero una constante igual a 0, para que los valores del segundo cluster ósea del periodo de muestreo, se trasformen y empiecen por 0, sin necesidad de especificar valores al resto. Por último se conectan los Data del subdiagrama Extrae de FIFO y el cluster resultante lo conectamos a un indicador grafico *Waveform Graph*. De esta manera se puede comprobar si los datos obtenidos son los que esperamos.

El último paso es como se hizo en la medición sin FIFO, salvando unas diferencias con respecto al tamaño del Array a crear.

### Creación Array de fecha, hora y datos

#### 1. Array de fecha y hora

Una vez que tenemos los Arrays 1D de fecha y hora, cablearemos a la función *Initialize Array* para crear un array de dimensiones las regidas por el número de lecturas de los datos.

Por lo que la salida de datos SGL, mediante *Array size* se obtiene el número de elementos que contiene e *Index Array* para acceder al valor de un elemento de ese Array se conecta a la función anterior.

Así tendremos un Array con los mismos valores de fecha/hora en todos los elementos.

#### 2. Array de datos

Se conecta los valores Data (SGL) a Number to fractional String y posteriormente se trasponen y se une junto con el Array de fecha/hora en la función *Build Array*. Importante es haciendo clic con el botón derecho sobre la función, activemos *Concatenate Inputs*, ya que así todas las entradas se anexionaran en orden, formando una matriz de salida de la misma dimensión que la entrada de mayor dimensión conectada.

Solo es necesario trasponer y representar en un indicador Array que sale por pantalla.

Finalmente se guardan los valores en un .txt igual que lo explicado anteriormente.



Programación flexible de controladores de automatización compactos comerciales para la monitorización de aerogeneradores residenciales

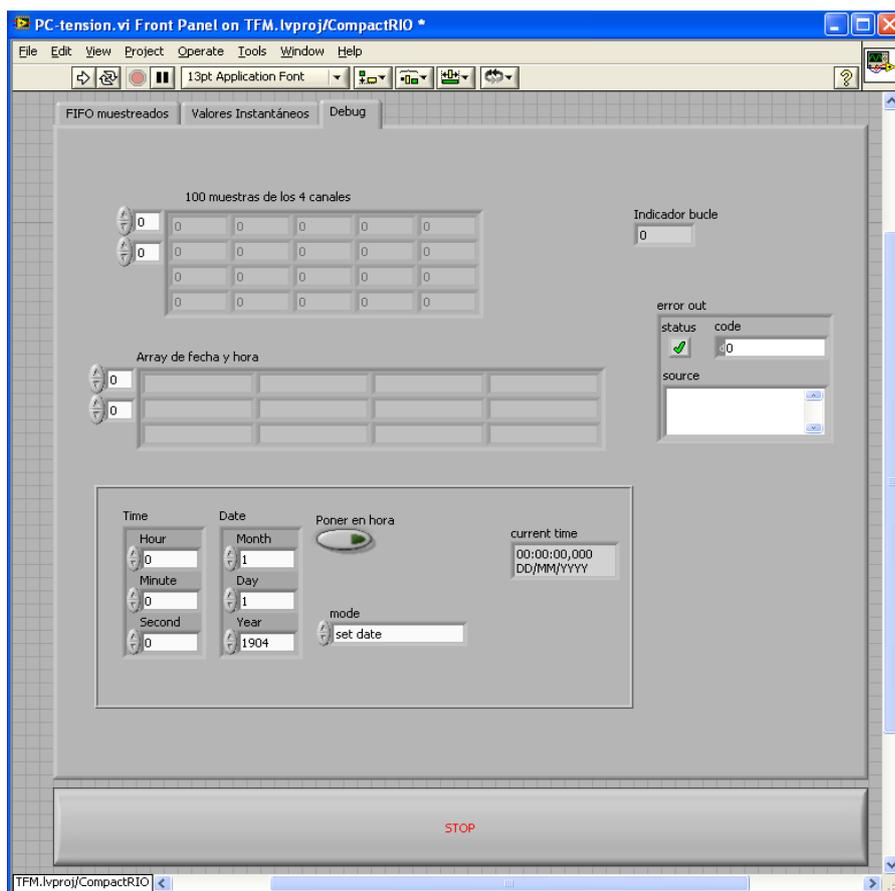
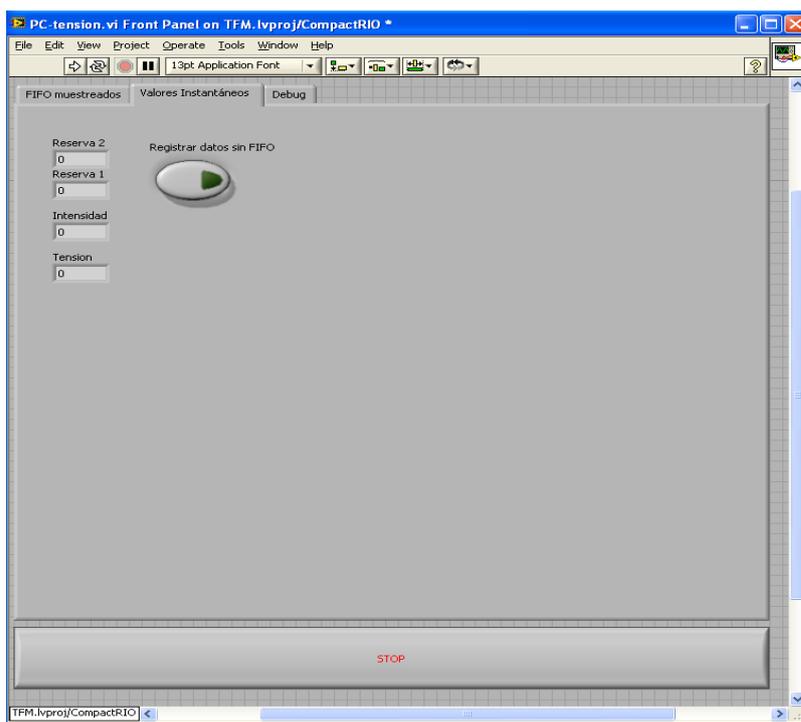


Figura 31: Programación Panel de control Bucle con FIFO.

## 5. Resultados

Una vez programado todo se presiona *Run* para obtener los resultados, que como ya se explico se accede a través del servidor utilizado.

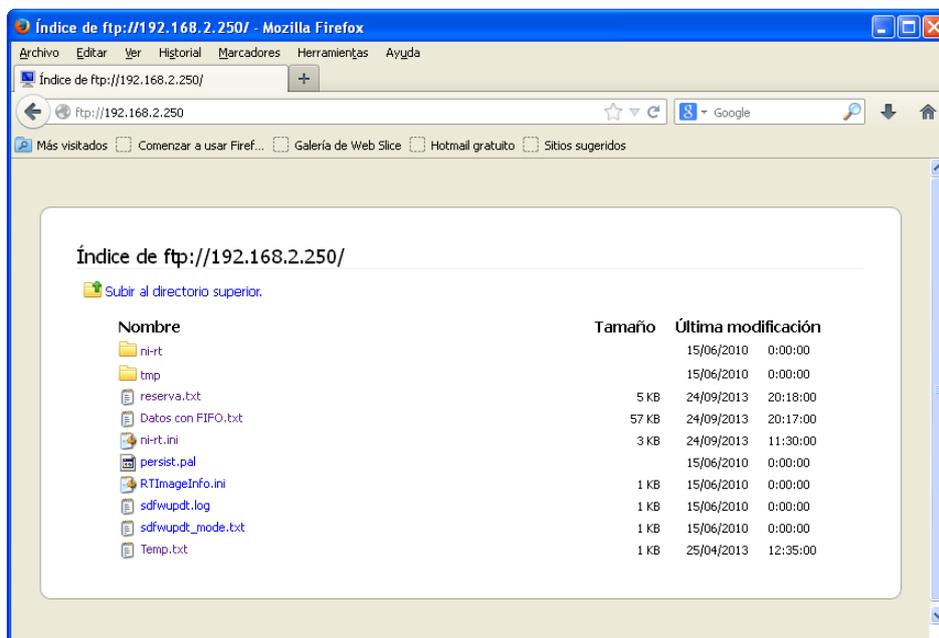


Figura 32: Acceso a los archivos de toma de datos.

### A. Toma de datos sin FIFO

A continuación se muestran los valores obtenidos de manera instantánea en LabVIEW y en el archivo .txt donde están todos los valores. Se comprueba que mide cada 6 seg y toma una lectura en los 4 canales, como lo habíamos programado.

24/09/2013	20:27:22	4.174774	0.000000	-6.092346	6.094452
24/09/2013	20:27:28	4.175415	0.000000	-4.230255	4.231781
24/09/2013	20:27:34	4.176056	0.000000	-0.055054	0.055603
24/09/2013	20:27:40	4.175110	0.000000	3.796661	-3.795990
24/09/2013	20:27:46	4.175110	0.000000	5.772400	-5.772797
24/09/2013	20:27:52	4.176056	0.000000	3.260132	-3.259064
24/09/2013	20:27:58	4.174469	0.000000	-0.125427	0.126068
24/09/2013	20:28:04	4.174133	0.000214	-3.366425	3.367706
24/09/2013	20:28:10	4.007935	0.000000	-5.123108	5.122925

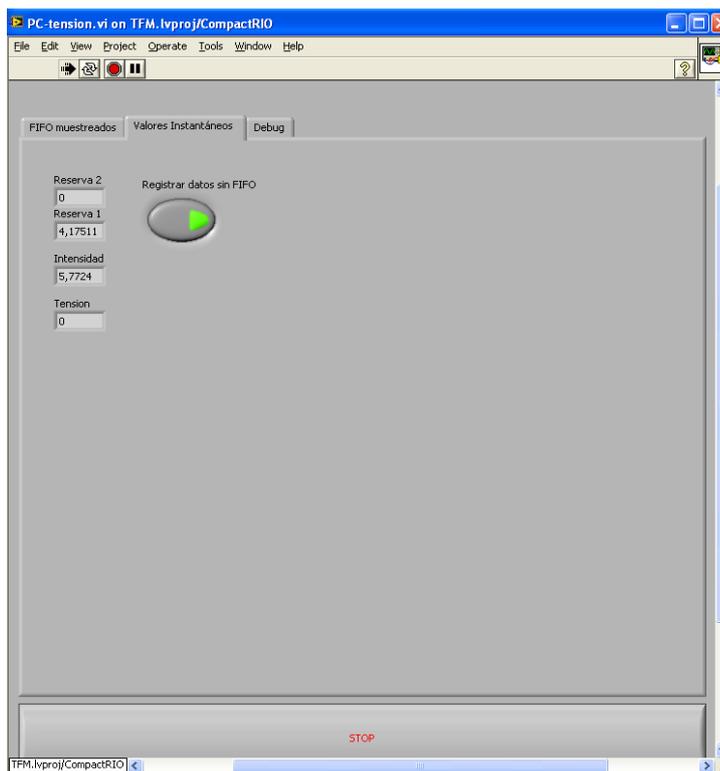


Figura 33: Resultados medición valores sin FIFO.

## B. Toma de datos con FIFO

CH1	24/09/2013	20:15:54	-3.167236	-3.138519	0.362640	3.772339	5.839691	5.713104	3.410767
-0.085968	-3.499451	-5.569366	-5.506561	-3.305908	0.158264	3.608765	5.786133	5.777832	5.777832
3.580078	0.120026	-3.333344	-5.513245	-5.565216	-3.473633	-0.051208	3.437866	5.721069	5.721069
5.835236	3.745544	0.324402	-3.166901	-5.449158	-5.613373	-3.632416	-0.250488	3.270782	3.270782
5.649994	5.881470	3.904968	0.528442	-2.995361	-5.376160	-5.656403	-3.790863	-0.457397	-0.457397
3.091278	5.569641	5.922913	4.064056	0.732819	-2.817444	-5.294861	-5.691162	-3.944550	-3.944550
-0.661469	2.909882	5.479401	5.956055	4.215851	0.935303	-2.639862	-5.203644	-5.719543	-5.719543
-4.092804	-0.866150	2.726227	5.381531	5.981903	4.361877	1.136169	-2.456848	-5.108643	-5.108643
-5.740906	-4.232483	-1.062256	2.544464	5.279816	6.002930	4.507263	1.340210	-2.268097	-2.268097
-5.002777	-5.758118	-4.373413	-1.265350	2.352844	5.166931	6.019836	4.647217	1.540771	1.540771
-2.077759	-4.891846	-5.770874	-4.508575	-1.466217	2.160583	5.048645	6.029724	4.779877	4.779877
1.739075	-1.887726	-4.775787							
CH2	24/09/2013	20:15:54	3.168610	3.138947	-0.360565	-3.771362	-5.837982	-5.712341	-3.409332
0.085724	3.500000	5.569763	5.507263	3.306854	-0.158234	-3.606934	-5.784943	-5.778015	-5.778015
-3.579468	-0.119110	3.334930	5.514221	5.565338	3.475067	0.050995	-3.437744	-5.720886	-5.720886
-5.833221	-3.744843	-0.323303	3.168274	5.449524	5.614899	3.632233	0.251099	-3.269226	-3.269226
-5.650177	-5.880585	-3.905792	-0.527496	2.996582	5.376923	5.656860	3.791290	0.458771	0.458771
-3.090881	-5.568115	-5.921600	-4.062653	-0.732025	2.818909	5.295197	5.692535	3.945313	3.945313
0.662354	-2.909424	-5.479126	-5.955048	-4.214478	-0.934631	2.640259	5.205872	5.720306	5.720306
4.093964	0.866547	-2.725739	-5.380951	-5.980927	-4.361847	-1.134735	-5.985855	5.108673	5.108673
5.742706	4.233154	1.062866	-2.543945	-5.279663	-6.001129	-4.507050	-1.339264	2.269104	2.269104
5.004181	5.760071	4.373932	1.266418	-2.352997	-5.166656	-6.017548	-4.646545	-1.539032	-1.539032
2.078491	4.892456	5.771118	4.509644	1.467163	-2.159515	-5.048950	-6.028900	-4.779419	-4.779419
-1.737244	1.888794	4.776642							
CH3	24/09/2013	20:15:54	4.174469	4.173523	4.174133	4.175110	4.174774	4.174133	4.175415
4.174133	4.175415	4.174133	4.174133	4.175110	4.175415	4.175415	4.173828	4.173828	4.173828
4.174774	4.175415	4.174774	4.174774	4.175110	4.174469	4.175110	4.175415	4.175415	4.175415
4.175415	4.175110	4.175415	4.174469	4.175110	4.172546	4.175110	4.174774	4.175110	4.175110
4.174774	4.175415	4.175110	4.174774	4.174133	4.174774	4.175110	4.175751	4.174133	4.174133
4.173828	4.176056	4.174774	4.172882	4.174774	4.174469	4.175751	4.174469	4.173828	4.173828
4.175324	4.174133	4.174133	4.173828	4.174774	4.175110	4.174469	4.173187	4.174774	4.174774
4.172882	4.175751	4.175415	4.175751	4.175110	4.174774	4.175110	4.174133	4.175110	4.175110
4.175415	4.174133	4.174774	4.174133	4.175110	4.175110	4.175751	4.174774	4.174469	4.174469
4.175110	4.174133	4.175751	4.174774	4.175110	4.175110	4.174774	4.174469	4.174133	4.174774
4.173187	4.173828	4.174133	4.174469	4.175110	4.172546	4.173187	4.174774	4.174133	4.174133
4.173187	4.174469	4.175415							
CH4	24/09/2013	20:15:54	-0.000732	-0.000732	0.000519	0.000214	-0.000427	0.000214	-0.000732
0.000214	-0.000732	-0.000092	-0.000092	-0.000092	-0.000427	-0.000092	0.000519	-0.000427	-0.000427
-0.000732	-0.000427	-0.000092	-0.000732	0.000214	0.000214	-0.000732	-0.000092	-0.000092	-0.000092
0.000214	-0.000427	-0.000732	0.000214	-0.000092	0.000854	-0.000427	-0.000092	-0.000092	-0.000092
-0.000427	-0.000427	-0.000427	-0.000092	-0.000427	-0.000092	-0.000427	-0.000427	-0.000092	-0.000092
-0.000427	-0.000092	-0.000092	-0.000092	0.000214	0.000214	-0.000427	-0.000427	-0.000427	-0.000427
0.000519	-0.000732	-0.000427	0.000214	-0.000427	-0.000732	-0.000427	-0.000427	-0.000092	-0.000092
0.000519	0.000214	-0.000427	-0.000427	-0.000732	-0.000732	-0.000092	-0.000092	-0.001038	-0.001038
-0.000427	-0.000732	0.000214	-0.000092	0.000214	0.000214	-0.000732	-0.000092	-0.000427	-0.000427
0.000519	-0.000092	-0.000092	-0.000732	-0.000732	-0.000092	0.000854	0.000214	-0.000092	-0.000092
0.000214	-0.000092	-0.000427	-0.000427	-0.000092	0.000214	-0.000427	0.000214	-0.000427	-0.000427

Programación flexible de controladores de automatización compactos comerciales para la monitorización de aerogeneradores residenciales

CH1	24/09/2013	20:16:54	-5.778839	-4.634857	-1.659424	1.969299	4.928772	6.035767	4.909637
1.934845	-1.694183	-4.653351	-5.778839	-4.758545	-1.854889	1.772552	4.799957	6.030029	
5.033020	2.128723	-1.499054	-4.524841	-5.770569	-4.874603	-2.043640	1.576477	4.667938	
6.018890	5.149719	2.318085	-1.303925	-4.394775	-5.759399	-4.985565	-2.231110	1.380707	
4.532135	6.005157	5.258453	2.501740	-1.107208	-4.261475	-5.743134	-5.088562	-2.413177	
1.185272	4.394379	5.985413	5.364624	2.688599	-0.908569	-4.120239	-5.722717	-5.187408	
-2.597473	0.985352	4.247711	5.959259	5.463135	2.871918	-0.706726	-3.973267	-5.694977	
-5.280182	-2.780487	0.777466	4.095642	5.927368	5.553040	3.052399	-0.503632	-3.822113	
-5.662140	-5.362762	-2.955841	0.575958	3.939728	5.888794	5.633728	3.224548	-0.304352	
-3.670990	-5.622284	-5.435760	-3.125763	0.376678	3.783478	5.843506	5.708313	3.397064	
-0.101593	-3.511902	-5.573822	-5.501770	-3.296051	0.172943	3.620239	5.789307	5.772095	
3.563477	0.099609	-3.351196							
CH2	24/09/2013	20:16:54	5.779327	4.635559	1.660309	-1.968597	-4.926788	-6.034271	-4.908173
-1.932617	1.695007	4.653870	5.780273	4.759583	1.855347	-1.771973	-4.799286	-6.029236	
-5.032532	-2.127655	1.499329	4.526978	5.772064	4.876068	2.044708	-1.576294	-4.667358	
-6.018188	-5.149897	-2.316711	1.304932	4.395691	5.760406	4.986847	2.232513	-1.379974	
-4.530701	-6.004303	-5.258514	-2.501007	1.108612	4.262512	5.743683	5.089081	2.414307	
-1.183655	-4.393738	-5.984406	-5.363922	-2.687866	0.909149	4.121124	5.723145	5.187866	
2.598297	-0.984192	-4.247284	-5.958527	-5.462708	-2.871246	0.707794	3.974365	5.696014	
5.280670	2.781342	-0.777466	-4.094849	-5.926331	-5.552032	-3.050812	0.504547	3.823181	
5.662567	5.363037	2.956512	-0.575165	-3.940826	-5.887207	-5.633453	-3.224396	0.305695	
3.671692	5.621857	5.436249	3.127258	-0.376007	-3.783356	-5.841766	-5.707611	-3.396088	
0.101807	3.513245	5.573883	5.502228	3.295807	-0.172119	-3.618591	-5.788727	-5.771362	
-3.561798	-0.098267	3.352295							
CH3	24/09/2013	20:16:54	4.174774	4.174469	4.175110	4.174469	4.175110	4.174774	4.175751
4.173187	4.174469	4.175110	4.175110	4.173187	4.174469	4.175110	4.175415	4.173523	
4.173523	4.174774	4.174469	4.175751	4.174469	4.175415	4.175110	4.174774	4.174774	
4.174774	4.175415	4.174469	4.174133	4.174774	4.175110	4.174133	4.174133	4.174774	
4.174774	4.174133	4.175415	4.174774	4.174133	4.174469	4.176056	4.174469	4.175110	
4.174469	4.175751	4.174469	4.174469	4.174774	4.175415	4.175751	4.174774	4.175110	
4.174469	4.173523	4.175110	4.175415	4.174774	4.174133	4.174774	4.175751	4.175415	
4.175110	4.175751	4.174469	4.175751	4.174774	4.175110	4.174774	4.174469	4.175415	
4.175110	4.176056	4.175110	4.174469	4.174133	4.175110	4.174774	4.175110	4.174469	
4.174774	4.175110	4.173828	4.176056	4.175751	4.175110	4.174133	4.176056	4.175110	
4.175110	4.174469	4.175110	4.176056	4.174774	4.175415	4.175751	4.175110	4.175110	
4.175415	4.174133	4.174774							
CH4	24/09/2013	20:16:54	-0.000427	-0.000427	-0.000092	-0.000092	-0.000427	-0.000092	-0.000092
-0.000092	0.000214	-0.000732	0.000214	-0.000427	-0.000427	-0.000427	-0.000092	-0.000427	
-0.000427	-0.000427	-0.000732	0.000214	-0.000427	-0.000427	-0.000427	-0.000092	-0.000732	
-0.000732	-0.000427	-0.000092	-0.000092	0.000214	-0.000732	-0.000427	-0.000092	0.000214	
-0.000427	-0.000732	-0.000427	-0.001038	-0.000092	-0.000427	-0.000092	-0.000092	-0.000427	
-0.000427	-0.000092	-0.000092	-0.000732	-0.000732	-0.000092	-0.000732	0.000214	-0.000092	
-0.000732	-0.000092	-0.000427	-0.000092	0.000214	-0.000427	-0.000427	0.000214	0.000519	
0.000214	-0.000427	0.000214	-0.000427	-0.000092	-0.000427	0.000519	-0.001038	0.000519	
-0.000427	0.000214	0.000214	-0.000092	0.000214	-0.000092	0.000214	-0.000427	-0.000092	
-0.000092	-0.000092	0.000214	0.000519	0.000214	-0.000092	-0.000092	-0.000092	0.000214	
0.000519	-0.000092	0.000214	0.000214	-0.000427	-0.000092	-0.000427	-0.000092	0.000214	
-0.000092	-0.000427	0.000519							

Se han tomado los valores de los cuatro canales en un instante inicial, grabándose 400 muestras y cuando el contador de iteraciones llega a 60, es decir que lleva 60 segundos midiendo toma otras 400 muestras. Por lo que se ha capturado esos dos instantes, para comprobar cómo pasa el tiempo que se ha querido y que los valores coinciden con los que aparecen en el Array del Panel de control, que solo muestra unas pocas por falta de espacio, pero si hacemos más grande esa matriz de resultados, saldrían los 400 valores.

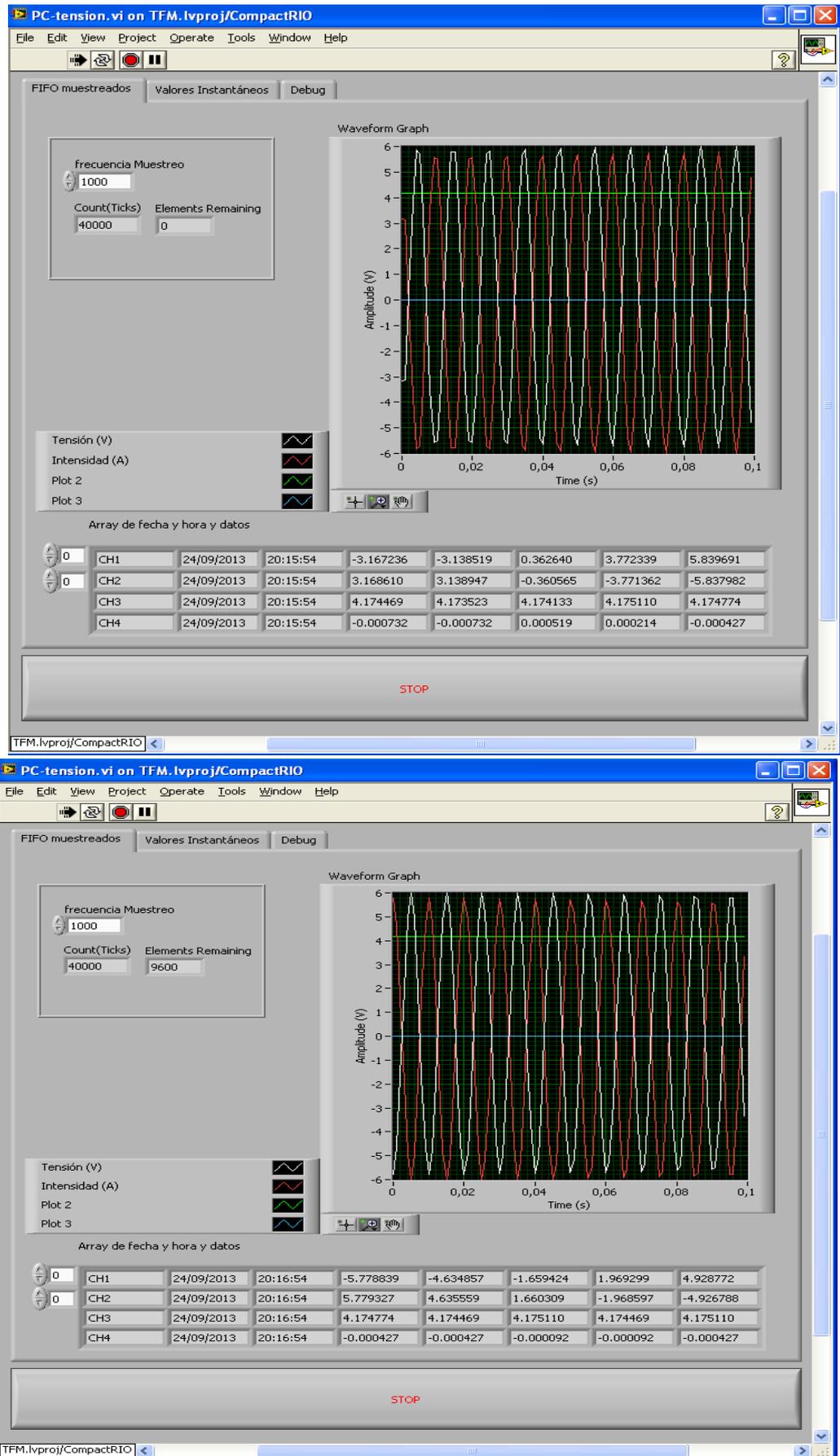


Figura 34: Resultados medición valores con FIFO.

## 6. Conclusiones y trabajos futuros.

Como hemos podido comprobar mediante un producto como LabVIEW unido a un controlador CompactRIO tenemos una capacidad muy grande en la adquisición y manejo de datos.

Hemos obtenido valores a una frecuencia muy grande mediante la herramienta FIFO, que permitirán poder realizar multitud de ensayos, ya no solo en aerogeneradores sino en cualquier dispositivo que se quieran medir magnitudes.

Gracias a la gran flexibilidad que ofrece el CompactRIO se podrá y se utilizarán en estudios posteriores para contrastar los datos en un banco de ensayo con los que se obtendrán en un emplazamiento real. Esto es posible una vez más a la gran versatilidad que ofrece el controlador, ya que se puede configurar para que las mediciones se vayan guardando mediante un dispositivo USB o volcándose directamente mediante una conexión inalámbrica.

La programación no tiene límites por lo que sobre éste programa se podría seguir modificando y aumentando su dificultad para obtener un modo tanto de obtención como de procesamiento más rápido y eficaz, ya que siempre lo que se busca es obtener el mejor resultado en el menor tiempo posible y con una eficacia lo mas alta posible.

## 7. Bibliografía

- Molina Martínez J.M. y Jiménez Buendía M. Programación grafica para Ingenieros. Marcombo, 2010.
- Programación Grafica LabVIEW para el desarrollo de aplicaciones SCADAS. Curso 60h 2010. Universidad Politécnica de Cartagena.