



industriales  
etsii

Escuela Técnica  
Superior  
de Ingeniería  
Industrial

# UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

## Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

TRABAJO FIN DE MASTER

MASTER EN ENERGÍAS RENOVABLES

**Autor:** Antonio Jose Fernández Agüera  
**Directores:** Angel Molina García  
Manuel Jiménez Buendía



Universidad  
Politécnica  
de Cartagena

Cartagena, Septiembre 2014

## **INDICE:**

1. INTRODUCCIÓN.....	7
1.1 EÓLICA DE PEQUEÑA POTENCIA.....	7
1.2 SITUACIÓN DE LA ENERGÍA MINI-EÓLICA.....	9
2. OBJETIVO.....	11
3. CONTROLADORES AUTOMÁTICOS PROGRAMABLES (PAC).....	12
3.1 GENERALIDADES.....	12
3.2 PACS ELIMINAN LA NECESIDAD DE HARDWARE A LA MEDIDA.....	13
3.3 FPGA FIELD PROGRAMABLE GATE ARRAY).....	14
3.3.1 DESCRIPCIÓN.....	14
3.3.2 VENTAJAS DE DISEÑAR SOBRE UNA FPGA.....	15
3.3.3 USO DE DMA FIFO.....	15
3.3.4 ALGUNAS FAMILIAS DE FPGAS COMERCIALES, UTILIZADAS EN EL MERCADO ACTUAL.....	16
3.4 LABVIEW PARA CONTROL.....	16
3.5 PACS DE NATIONAL INSTRUMENTS.....	18
3.5.1 HARDWARE DE ADQUISICIÓN NI SINGLE-BOARD RIO.....	19
3.5.2 DISTRIBUCIÓN MÁS RÁPIDA CON LOS SISTEMAS EMBEBIDOS CON LABVIEW.....	20
3.5.3 EL MÓDULO LABVIEW REAL-TIME.....	21
3.5.4 MÓDULO LABVIEW FPGA.....	21
3.5.5 HERRAMIENTAS DE DRIVERS DE CONEXIÓN DE PROGRAMAS Y APLICACIONES.....	22
3.5.6 TRANSICIÓN DESDE LA CREACIÓN DEL PROTOTIPO A LA IMPLANTACIÓN.....	23
3.5.7 ALTO RENDIMIENTO CON LA PROGRAMACIÓN CON EL SIMPLE CICLO WHILE LOOP (SCLP).....	23
4. COMPARATIVA CON PLCS TRADICIONALES.....	29
5. EQUIPO EMPLEADO.....	31
5.1 DESCRIPCIÓN.....	31
6. METODOLOGÍA.....	34
6.1 CONFIGURACIONES INICIALES.....	34
6.1.1 CONFIGURACIÓN WIFI SOBRE MYRIO.....	34
6.1.2 CONFIGURACIÓN DÍA Y HORA EN MYRIO.....	35
6.2 SIMULACIÓN DE VARIABLES A MEDIR.....	36
6.3 CONFIGURACIÓN DISPOSITIVO.....	37
6.4 PROGRAMACIÓN ARCHIVOS VI DE LA TARJETA DE ADQUISICIÓN DE DATOS.....	39
6.4.1 DIAGRAMA DE BLOQUES.....	39
6.4.2 PANEL DE CONTROL.....	43
6.4.3 COMPILACIÓN.....	44
6.5 PROGRAMACIÓN ARCHIVOS VI DEL CONTROLADOR COMPACTRIO.....	44

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

6.5.1	DIAGRAMA DE BLOQUES.....	44
6.5.2	PANEL FRONTAL.....	52
6.6	COMUNICACIONES.....	54
6.6.1	PUBLICAR EL PANEL FRONTAL EN LA WEB. ....	54
6.6.2	COMUNICACIÓN MEDIANTE UN CORREO ELECTRÓNICO.....	57
7.	RESULTADOS.....	59
7.1	PREPARACIÓN ENSAYO .....	59
7.2	RESULTADOS.....	59
8.	CONCLUSIONES Y TRABAJOS FUTUROS. ....	62
9.	BIBLIOGRAFÍA. ....	63

**FIGURAS:**

- Figura 1. Clasificación de los controladores, función de la velocidad de procesado y la memoria.
- Figura 2: Arquitectura demostrativa, no detallada de una FPGA de Xilinx
- Figura 3. Clasificación de equipos RIO.
- Figura 4: NI Single- Board RIO integra un procesador, una FPGA y E/S analógicas y digitales en una sola tarjeta que se puede programar con las herramientas gráficas de LabVIEW.
- Figura 5: Código de Labview FPGA que sintetiza los circuitos y ejecuta secuencialmente cuando existen dependencia de datos.
- Figura 6: El código LabVIEW FPGA utiliza registros en cada función para enganchar los datos en cada ciclo del reloj de nivel superior, normalmente de 40 MHz (período de 25 ns). Aquí, los registros de hardware están representados por cajas etiquetadas "R."
  
- Figura 7: El SCTL es una estructura de bucle que especifica el reloj para ser utilizado por el código y asume una latencia máxima iteración de un ciclo de reloj.
- Figura 8: El SCTL representa un circuito síncrono donde las señales fluyen a través y están enclavadas sólo en puntos específicos como el de E / S, los controles y los indicadores y otras construcciones. El SCTL especifica el reloj utilizado para conducir sus contenidos y las señales deben propagarse de entre registros en menos de un ciclo de reloj, que es de 25 ns para este ejemplo
- Figura 9: Estructura procesador Compact RIO
- Figura 10: Equipo empleado en los ensayos. myRIO.
- Figura 11: Partes del Myrio
- Figura 12: Estructura procesador MyRIO
- Figura 13: Señales entrada y salida MyRIO
- Figura 14: Composición del MyRIO multiplexados entradas/salidas analógicas.
- Figura 15: Esquema Configuración WEB del MyRIO.
- Figura 16: Esquema Configuración de las redes de conexión del MyRIO.
- Figura 17: Esquema Configuración Horario del MyRIO.
- Figura 18: Esquema Montaje simulación señales

- Figura 19: Configurar el Hardware MyRIO en MAX
- Figura 20: Entrada a Labview 2013.
- Figura 21: Esquema de conexión a MyRIO mediante USB
- Figura 22: Esquema Carpetas Proyecto
- Figura 23: Esquema conversión voltaje de
- Figura 24: Propiedades del tipo de dato.
- Figura 25: Características de la FIFO
- Figura 26: Datos de la FIFO.
- Figura 27: Diagrama de bloque VI de la FPGA.
- Figura 28: Panel de control tarjetas de datos de la FPGA.
- Figura 29: Compilación de la FPGA
- Figura 30: Insertar entrada FPGA
- Figura 31: Configurar la FPGA que debe leer.
- Figura 32: Creación del Loop Timer
- Figura 33: Configuración tiempo de descarga de datos de la FIFO.
- Figura 34: Configuración Control Read de la FIFO.
- Figura 35: Configuración de la FIFO Read.
- Figura 36: Diagrama de Bloque del Extraer de la FIFO.
- Figura 37: Configuración función Bundle para hacer un gráfico de los datos.
- Figura 38: Creación de Array de fecha y hora.
- Figura 39: Panel de control del SCADA.
- Figura 40: Panel de control secundario del SCADA.
- Figura 41: Configuración Web Server
- Figura 42: Opciones Control Web Publishing
- Figura 43: Nombre del Control Web Publishing
- Figura 44: Configuración de ubicación Web
- Figura 45: Esquema Data communication
- Figura 46: Insertar función enviar email
- Figura 47: Configurar correo email
- Figura 46: Insertar función enviar email

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

- Figura 47: Comprobación de correo.
- Figura 48: Vista parámetros de medida
- Figura 49: Ubicación de los datos de medida.
- Figura 50: Vista parámetros de medida en panel frontal principal
- Figura 51: Vista parámetros de medida en panel frontal principal

## 1. Introducción.

Actualmente, las energías renovables están en continuo desarrollo y se encuentran inmersas en un proceso de integración dentro del sector eléctrico. Entre ellas, las que más han evolucionado desde el punto de vista de instalaciones conectadas a la red son las fuentes eólica y fotovoltaica. Mientras que las instalaciones fotovoltaicas es posible encontrarlas en un amplio abanico de potencias, las instalaciones eólicas se han centrado principalmente en parques de elevada potencia, estando en la actualidad las aplicaciones de mini-eólica en un proceso de desarrollo y de paulatina extensión hacia los sectores terciario y residencial. Existen diversos aspectos que dificultan su generalización como fuente alternativa en aplicaciones de baja potencia, como son su pequeño reporte energético y las limitaciones económicas en comparación con instalaciones de mayor capacidad de potencia.

Dentro de este escenario, el presente trabajo se centra en el desarrollo y aplicación de **una herramienta de monitorización de bajo coste y sencilla de programar** que nos permita integrar su uso en el uso residencial y que nos permita hacer una conexión a red.

Dentro del mercado actual de aplicaciones comerciales, existen diversos fabricantes y software de medida de energía y monitorización que permiten tener acceso remoto y almacenamiento de datos. Éstos normalmente presentan una mayor facilidad de conexión con equipos o instalaciones ligadas a la misma casa comercial, o bien ofrecen características de almacenamiento y programación limitadas y de alto coste de extensión, lo cual las hace difícilmente aplicables al amplio sector industrial y las aleja de la mayoría de pequeñas industrias. Por ello, se pretende en este trabajo el desarrollo de una aplicación de bajo coste y alta flexibilidad de programación que permita la adquisición de variables eléctricas en un rango de muestreo del orden de kHz, lo cual añade una mayor versatilidad a los datos almacenados y permite al usuario un extenso abanico de tratamiento de datos posterior. En concreto, el trabajo se basará en la instalación de un controlador de un precio económico pero con prestaciones suficientes para las características deseadas, reduciendo las necesidades de equipamiento al empleo exclusivamente de tarjetas SingleBoard RIO.

### 1.1 Eólica de pequeña potencia.

Los parques eólicos de gran potencia son fundamentales para aumentar la contribución de la energía de origen renovable en el sistema eléctrico nacional. Sin embargo, todavía no se ha aprovechado en España la **capacidad de la tecnología eólica para aportar energía renovable de forma distribuida,**

mediante su integración en entornos urbanos, semi-urbanos, industriales y agrícolas, especialmente **asociada a puntos de consumo de la red de distribución**.

Las instalaciones eólicas de pequeña potencia presentan unas características propias, que las dotan de una serie de **ventajas adicionales respecto a la gran eólica**, como una potencial mayor eficiencia global por las pérdidas evitadas en las redes de transporte y distribución, y que permiten la integración de generación renovable sin necesidad de crear nuevas infraestructuras eléctricas. Además, pueden fomentar la implicación ciudadana en la mejora de la eficiencia energética, el autoabastecimiento energético y la lucha contra el cambio climático. A continuación, se citan las más significativas:

- Generación de energía próxima a los puntos de consumo.
- Versatilidad de aplicaciones y ubicaciones, ligado al autoconsumo, con posibilidad de integración en sistemas híbridos.
- Accesibilidad tecnológica al usuario final, facilidad de transporte de equipamientos y montaje.
- Funcionamiento con vientos moderados, sin requerir complejos estudios de viabilidad.
- Aprovechamiento de pequeños emplazamientos o de terrenos con orografías complejas.
- Suministro de electricidad en lugares aislados y alejados de la red eléctrica.
- Optimización del aprovechamiento de las infraestructuras eléctricas de distribución existentes.
- Bajo coste de operación y mantenimiento y elevada fiabilidad.
- Reducido impacto ambiental, por menor tamaño e impacto visual, y por su integración en entornos con actividad humana.

Técnicamente, estas aeroturbinas tienen una estructura similar a las grandes, solo que su diseño es mucho más simple (sistemas de orientación pasivos, generadores eléctricos robustos de bajo mantenimiento, ausencia de multiplicadores...). Su sencillez de funcionamiento hace que, en general, estas pequeñas instalaciones puedan ser atendidas por los propios usuarios.

En la actualidad en España, los pequeños aerogeneradores son sobre todo utilizados para el autoconsumo de edificaciones aisladas. Además, suelen ir acompañados de paneles solares fotovoltaicos formando parte de pequeños sistemas híbridos que, por medio de la combinación de la energía del sol y el viento, permiten garantizar el suministro eléctrico. Estos sistemas, bastante



fiables, incluyen unas baterías donde se almacena la energía sobrante para cuando no haya viento ni sol.

Otra posibilidad consiste en utilizar estas máquinas para producir energía y verterla a la red eléctrica. Esta opción todavía está muy poco desarrollada en España, si bien esto podría cambiar en esta década con unas condiciones más favorables

## **1.2 Situación de la energía mini-eólica.**

Las empresas vinculadas al sector de la **energía minieólica** han decidido apostar por **volcarse** en el **extranjero** ante el bloqueo que sufren en España, con normativas que impiden su desarrollo.

La Asociación de Productores de Energía Renovables (sección minieólica) y el CIEMAT han firmado un acuerdo de colaboración para impulsar la tecnología **minieólica española** en los foros y mercados internacionales en los que ambas instituciones participan.

Así lo harán, por ejemplo, en la Agencia Internacional de la Energía (International Energy Agency-IEA) o en la Asociación Mundial de Energía Eólica (World Wind Energy Association-WWEA). El sector minieólico español vive una situación de crisis por la falta de apoyo y de políticas institucionales que fomenten su desarrollo para cumplir los objetivos previstos en el Plan de Energías Renovables 2011-2020, que prevé integrar 300 MW en el sistema eléctrico español.

El bloqueo del Real Decreto sobre autoconsumo añade más incertidumbre al sector, que ve casi imposible que con las actuales circunstancias cambien las limitaciones existentes en el mercado minieólico en España. Ante esta situación de estancamiento que vive la minieólica en nuestro país, los agentes del sector reunidos en torno a APPA Minieólica han decidido aunar fuerzas para fomentar en los mercados exteriores el potencial y la calidad de la tecnología minieólica española.

Para ello, han puesto en marcha una campaña que promoverá la tecnología minieólica bajo la marca SmallWindSpain, que mostrará en el ámbito internacional que España es uno de los países con mejores tecnólogos en esta disciplina y dará visibilidad al tejido industrial que lidera esta iniciativa.

Empresas como Baiwind, Bornay, Ennera o Kliux Energies disponen de aerogeneradores de pequeña potencia de primerísimo nivel que pueden competir perfectamente en el mercado mundial.

Como primera acción de la campaña, el pasado mes de marzo los industriales minieólicos españoles asistieron de manera conjunta a la feria New Energy Husum 2014, en la que se incluye el mayor evento internacional sobre

tecnología minieólica: la 5ª edición de la Cumbre Mundial sobre Eólica de Pequeña Potencia (5th World Summit for Small Wind 2014-WSSW, organizada anualmente por la WWEA.

La delegación, encabezada por APPA Minieólica, contó con un stand propio bajo la marca SmallWind Spain. Durante los cuatro días que duró la Feria, los tecnólogos españoles participaron como ponentes en distintas conferencias y dieron a conocer el potencial del sector minieólico español.

Asimismo, durante la celebración de esta Cumbre Minieólica fue presentada la primera edición del Catálogo de Aerogeneradores de Pequeña Potencia, con información detallada de las características técnicas de cada uno de los modelos diseñados, fabricados y comercializados por los miembros de APPA Minieólica.

A día de hoy figuran en el mismo cuatro fabricantes (Baiwind, Bornay, Ennera y Kliux Energies), aunque se prevé incluir en él a todos los que tengan interés en conseguir mayor visibilidad internacional.

APPA Minieólica ha iniciado una campaña para tratar de aunar en torno a esta iniciativa a todos los agentes del sector para que se integren, participen y ayuden a fortalecer la imagen del sector minieólico español, con grandes capacidades para convertirse en referente mundial.

En esta línea se enmarca el reciente acuerdo de colaboración firmado con CIEMAT, que sin duda impulsará y dará visibilidad internacional a la industria minieólica española. Vista la buena acogida que ha tenido esta primera iniciativa, ya se está trabajando en llevar a cabo nuevas acciones y concretar así algunas de las cuestiones más avanzadas en las que el sector minieólico lleva tiempo trabajando pero que no han tenido el apoyo institucional necesario para salir adelante.

## **2. Objetivo.**

El objetivo principal del presente Trabajo Fin de Máster, dentro del Máster en Energías Renovables impartido en la Universidad Politécnica de Cartagena, se centra en el desarrollo de un SCADA (Supervisory Control And Data Acquisition) mediante el uso de Labview y programación de una FPGA para registrar y monitorizar las variables eléctricas y de funcionamiento de un aerogenerador de baja potencia con implementación dentro del sector residencial. Como valor añadido a los objetivos a desarrollar, indicar que el equipamiento de monitorización es de aplicación en el sector industrial, lo cual facilita el acercamiento del alumno a los equipos y dispositivos de uso actual en dicho sector.

La aplicación a desarrollar permitirá al usuario parametrizar la toma de datos según su necesidad, sin tener que depender de las limitaciones existentes en los equipos ya preconfigurados de monitorización y comercializados en la actualidad.

### 3. Controladores Automáticos Programables (PAC).

#### 3.1 Generalidades.

El PAC (Programmable Automation Controller) es utilizado para describir una nueva generación de controladores industriales que combinan la funcionalidad PLC y PC. El acrónimo PAC es utilizado por vendedores tradicionales PLC para describir los sistemas de alto desempeño.

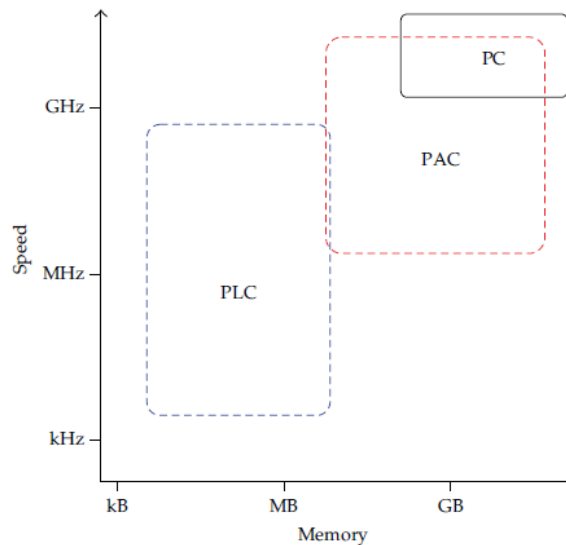


Figura 1. Clasificación de los controladores, función de la velocidad de procesamiento y la memoria.

Como podemos observar en la figura 1, los PAC son una mezcla entre PC y PLC, con velocidad del proceso y una memoria intermedia.

La Asociación ARC (Corporación de Investigación de Automatización) identifica cinco características PAC esenciales. Estos criterios caracterizan la funcionalidad del controlador al definir las capacidades del software:

- **Funcionalidad Multi Dominio.** Al menos 2 plataformas sencillas de proceso lógico, en movimiento, control PID y programas. Exceptuando algunas variaciones en E/S para señalar protocolos específicos como SERCOS; lógica, movimiento, proceso, y PID son simplemente una función del software. Por ejemplo, el control de movimiento es un software de control en ciclos que lee entradas digitales de un codificador de cuadratura, desempeña ciclos de control análogo, y entrega una salida de señal análoga para controlar un programa.
- **Plataforma de desarrollo multi disciplinaria sencilla** incorporando etiquetado común y una sola base de datos para tener acceso a todos los parámetros y funciones. Debido a que los PACs están diseñados para aplicaciones más avanzadas como los diseños multidominios, requieren

de software mucho más avanzado. Para que el diseño del sistema sea eficiente, el software debe estar integrado en un paquete en vez de presentar distintas herramientas de software las cuales no están creadas para trabajar entre si.

- **Herramientas de software que permiten el diseño del proceso de flujo a través de varias máquinas o unidades de proceso**, en conjunto con IEC61131-3, guía al usuario, y administración de datos. Otro componente que simplifica el diseño del sistema son las herramientas de desarrollo gráfico de alto nivel que ayudan a traducir el concepto del proceso de ingeniería al código que actualmente controla la máquina.
- **Arquitecturas modulares**, abiertas esas aplicaciones que reflejan la industria desde planos de máquinas en la industria a operaciones unitarias en plantas de proceso. Debido a que todas las aplicaciones industriales requieren configuración significativa, el hardware debe ofrecer modularidad para que el ingeniero pueda escoger y elegir los componentes apropiados. El software debe habilitar que el ingeniero adicione y remueva módulos para diseñar el sistema requerido.
- **Emplear estándares reales para interfases y lenguajes de red**, etc., como búsquedas TCPIP, OPC & XML, y SQL. La comunicación con las redes empresariales son críticas para los sistemas de control modernos. Aunque los PACs incluyen un puerto Ethernet, el software para comunicación es la clave para evitar conflictos de integración con el resto de la planta.

### **3.2 PACs Eliminan la Necesidad de Hardware a la Medida**

Aunque los PACs representan los controladores programables actuales, los PACs futuros serán moldeables para poderse adaptar e incorporar a la tecnología. Un ejemplo es la habilidad para utilizar software y definir así el hardware. Los Arreglos de Compuerta Programables de Campo (FPGAs) son componentes electrónicos comúnmente utilizados por fabricantes electrónicos para crear chips a la medida, permitiendo la inteligencia en nuevos dispositivos. Estos dispositivos consisten en bloques de lógica configurables con la habilidad de desempeñar una variedad de funciones, la interconexión programable que actúa en los interruptores para conectar las funciones de bloques, y los bloques E/S que pasan datos dentro y fuera del chip. Al definir la funcionalidad de los bloques de lógica configurables y la manera en las cuales se encuentran conectados unos a otros y el E/S, los diseñadores de electrónicos pueden crear chips sin sacrificar la producción de ASIC a la medida. Los FPGAs son comparables a tener una computadora que literalmente conecta nuevamente los circuitos internos para ejecutar una aplicación en específico.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

La tecnología FPGA ha estado disponible solamente para diseñadores de hardware los cuales eran altamente eficientes en lenguajes de programación de bajo nivel como los VHDL. Sin embargo, los ingenieros de control, hoy en día, pueden utilizar los FPGA de LabVIEW para crear algoritmos de control a la medida los cuales son descargados a los chips FPGA. Esta capacidad permite a los ingenieros incorporar funciones al hardware extremadamente críticas en tiempo como el límite y proximidad de detección del sensor y monitoreo de la salud del sensor. Debido a que el código de control se ejecuta directamente en silicón, es posible que los ingenieros creen aplicaciones rápidamente que incorporan protocolos de comunicación a la medida o ciclos de control de alta velocidad: hasta ciclos de control digital de 1 MHz y ciclos de control análogo de 200 kHz.

### 3.3 FPGA Field Programmable Gate Array).

#### 3.3.1 Descripción.

Una FPGA (Field Programmable Gate Array o lo que es lo mismo Arreglo d Compuertas Programables en Campo). La FPGA se compone de recursos no comprometidos que pueden ser seleccionados, configurados e interconectados por el usuario.

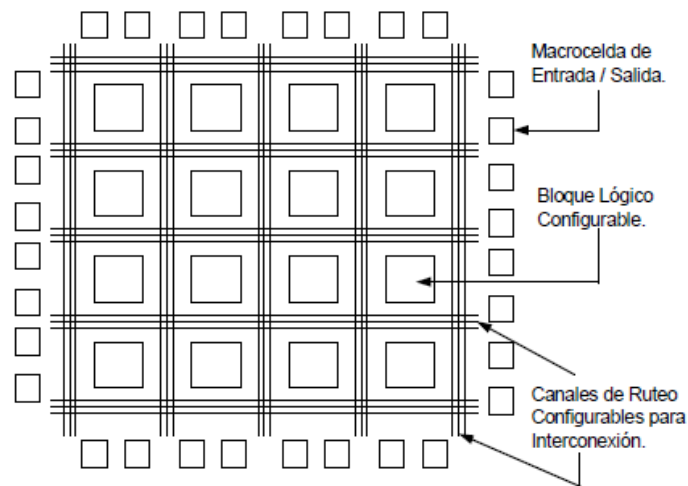


Figura 2: Arquitectura demostrativa, no detallada de una FPGA de Xilinx

Los módulos FPGA, se interconectan por medio de Canales Configurables.

Estos dispositivos se componen de ciertos números de módulos Lógicos que determinan la capacidad del dispositivo. Los módulos son independientes entre si y pueden interconectarse para formar un módulo más complejo. Dependiendo de los fabricantes estos módulos pueden ser bloques configurables como es el caso de las FPGAs de la marca Xilinx, concretamente

la que dispone nuestro controlador, o bien elementos de función fija formados por arreglos de compuertas, como ocurre es los dispositivos de Actel.

Al proceso de interconexión se le llama enrutamiento y consiste en determinar la mejor estrategia de interconectar los módulos, ya sea en forma manual o mediante alguna herramienta de diseño gráfico (EDA) en nuestro caso uso de Labview.

Comentar que los dispositivos de Xilinx y Actel son los más utilizados comercialmente.

### **3.3.2 Ventajas de diseñar sobre una FPGA.**

- Minimización del número de componentes en un diseño. Con esto se reducen los gastos de inventario, inspección y prueba, así como el número de fallos a nivel circuito impreso, propiciando un ahorro de espacio físico. Así, una medida de la eficiencia de un Dispositivo Programable se expresa mediante el número de dispositivos de función fija (Circuitos Integrados de Catálogo) que pueden reemplazarse.
- Reducción en el tiempo de diseño. Debido a su naturaleza programable, reducen el tiempo y los costos de desarrollo, no sólo de nuevos productos sino también de aquellos que requieren modificaciones (reingeniería), ya que son reutilizables tantas veces como sea necesario. Esto último se debe a que los cambios en el diseño son realizados mediante una nueva programación que se prueba inmediatamente si se está utilizando un dispositivo programable en el mismo circuito (IS).
- Uso de una gran variedad de herramientas de Diseño Asistido por Computadora (CAD), disponibles actualmente en el mercado. Estas herramientas promueven y facilitan el diseño sobre este tipo de dispositivos. Así mismo, no se requiere de grandes recursos de cómputo

### **3.3.3 Uso de DMA FIFO.**

Esta aplicación permite desarrollar aplicaciones de alta velocidad de almacenamiento de adquisición de datos en los dispositivos RIO. La FPGA permite uso de la RAM del host como si fuera suya. Se ofrece así mejoras en el rendimiento respecto a las FIFO tradicionales.

La transferencia de DMA usa una arquitectura FIFO, compuesta de 2 partes. Un bloque está en el dispositivo y utiliza la RAM de este, y la segunda parte está en la máquina huésped, usando la memoria de este.

El DMA se transfiere automáticamente los datos de la RAM del dispositivo a la RAM del host o huésped.

### **3.3.4 Algunas Familias de FPGAs Comerciales, utilizadas en el mercado actual.**

Los diseñadores coinciden en afirmar que desde el punto de vista usuario, existen tres fabricantes mayoritarios en la distribución de FPGAs y software de soporte: Xilinx, Altera y Actel. En el mercado mundial podemos encontrar otros tantos con producciones menores pero que figuran también como FPGAs útiles: Lucent, Texas Instruments, Philips, QuickLogic, Cypress, Atmel, etc.

Todo fabricante ofrece la información de sus productos por Internet, donde podemos encontrar hojas de especificaciones, notas de aplicación (proyectos realizados) y tutoriales para el manejo de dispositivos y software de diseño, entre otras opciones. Se recomienda visitar los sitios WEB para familiarizarse con los términos y facilitar la búsqueda de información. En cada sitio WEB, es posible solicitar una copia gratuita de los Data Books (manuales), así como una copia de evaluación del software de diseño. Por lo general, en el sitio WEB aparece la opción "PRODUCTS" (productos), con la que accedemos a las Digital Libraries (Bibliotecas Digitales 5) que son los manuales que contienen la información técnica de cada familia disponible.

El desarrollo de sistemas en FPGAs en tan amplio y necesario en la actualidad, que la evolución en las técnicas y metodologías en diseño digital electrónico, ha hecho que en la mayoría de las Universidades adopten la enseñanza de estos conceptos como parte de la formación académica del profesionalista en áreas afines. Cada fabricante tiene instituido un University Program (Programa Universitario) en el que donan a las Universidades, software de diseño y hardware (tarjetas de programación y dispositivos) tras una solicitud formal que justifique el uso de los recursos con propósito docente. En los mismos sitios WEB está la información pertinente.

Las direcciones de los sitios son:

- Xilinx: <http://www.xilinx.com>
- Altera: <http://www.altera.com>
- Actel: <http://www.actel.com>

### **3.4 LabVIEW para Control.**

Seleccionamos la plataforma CompactRIO y LabVIEW por su capacidad de adaptación a nuestras necesidades concretas: adquisición de señales a alta frecuencia, transmisión de datos por una red local hacia el SCADA vía OPC (OLE for Process Control estándar de comunicación en el campo del control y supervisión de procesos industriales, basado en una tecnología Microsoft), capacidad de almacenamiento de datos. LabVIEW se caracteriza por dar una



visión de integración y programación más rápida de lo común en este tipo de entornos.

Debido a las características de LabVIEW y la facilidad de uso de la programación gráfica, LabVIEW basado en PACs es adecuado para aplicaciones que requieren:

- **Gráficos.** Debido a que el programador de LabVIEW nativamente construye una interfase de usuario, puede incorporar gráficos y sistemas de control HMI fácilmente.
- **Mediciones** (adquisición de datos, visión y movimiento de alta velocidad). National Instruments tiene una fuerte historia en E/S de alta velocidad, incluyendo adquisición de visión, para que pueda incorporar mediciones como vibración y visión de máquina a sus sistemas de control estándar.
- **Capacidades de Procesamiento.** En algunas aplicaciones, requiere de algoritmos de control especializados, procesamiento de señales avanzadas, o acceso a datos. Utilizando LabVIEW, usted puede incorporar códigos de control a la medida construidos utilizando herramientas de NI o de terceros, implementar procesamiento de señales como los JTFA, o acceso de datos locales o remotos.
- **Plataformas.** Con LabVIEW, usted puede crear códigos que ejecuten gran variedad de plataformas incluyendo una PC, un controlador incluido, un chip FPGA, o un PDA manual.
- **Comunicación.** LabVIEW hace fácil la transferencia de datos a la empresa con herramientas como conectividad de base de datos, OPCs, e interfases de operador vía web.

Además, el entorno de programación LabVIEW de NI es lo último en software de diseño de sistemas usado por ingenieros y científicos para diseñar, generar prototipos y desplegar aplicaciones embebidas de control y monitoreo de manera eficiente. Combina cientos de bibliotecas pre-escritas, estrecha integración con hardware comercial y una variedad de enfoques de programación incluyendo desarrollo gráfico, scripts de archivos .m y conectividad a código ANSI C y HDL existente. Ya sea para diseñar dispositivos médicos o robots complejos, usted puede reducir el tiempo al mercado y los costos totales del diseño monitoreo y control embebidos con LabVIEW.

El empleo de Labview para desarrollo de sistemas embebidos, se puede resumir en 10 razones, que son las siguientes:

1. Genera Prototipos Más Rápido con un Entorno de Diseño Gráfico.
2. Reutiliza Fácilmente Código Embebido e IP Existente.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

3. Máxima Flexibilidad con una Arquitectura en Tiempo Real y Basada en FPGA.
4. Realiza Iteraciones rápidamente con Habilidades Integradas de Depuración, Simulación e Interfaz de Usuario.
5. Integra Cientos de Análisis Avanzado u Otro IP
6. Implementa Lógica de Control FPGA sin Aprender HDL.
7. Empieza a Trabajar Inmediatamente con Ejemplos para Abrir y Ejecutar.
8. Fácil Despliegue a Plataformas Optimizadas por Volumen al Mantener su Inversión de Software.
9. Plataforma de Software Escalable y Flexible.
10. Colabora y Desarrolla con una Comunidad de Ingenieros en Todo el Mundo.

### **3.5 PACs de National Instruments.**

National Instruments ofrece cinco plataformas basadas en PAC de LabVIEW (ver figura 3):

- PXI
- Compact FieldPoint
- Compact Visión System
- PCs industriales
- CompactRIO

La que se va a emplear en nuestro proyecto es la de **CompactRIO**, en concreto el dispositivo **MyRIO**.

National Instruments ofrece diversos hardware de propósito general disponible comercialmente COTS (Comercial off-the-shelf) que comparte un arquitectura común reconfigurable de E/S (RIO). Esta arquitectura combina un procesador en tiempo real, una FPGA (Field-Programmable Gate Array) y una amplia gama de E/S, incluyendo E/S analógicas, digitales, de movimiento y de comunicación.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

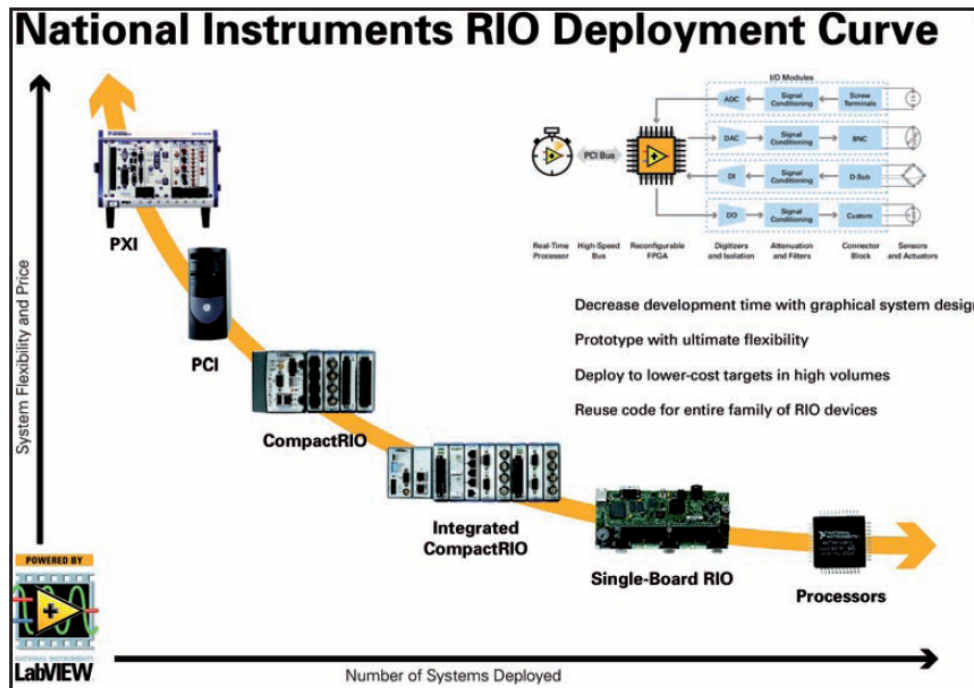


Figura 3. Clasificación de equipos RIO.

Usando esta arquitectura estándar y las herramientas gráficas de desarrollo de NI LabVIEW se pueden diseñar y crear rápidamente prototipos de sistemas embebidos, máquinas de control y monitorización industrial flexibles y dotadas de hardware de alto rendimiento.

Gracias a la capacidad de reutilización completa del código, se puede convertir el prototipo en un sistema implantado con un coste optimizado utilizando la misma arquitectura de hardware para reducir costes y el tiempo de lanzamiento al mercado.

### 3.5.1 Hardware de adquisición NI Single-Board RIO.

Los nuevos productos NI Single- Board RIO amplían la familia de opciones de implantación de NI RIO al hardware embebido a nivel de tarjeta y de bajo costo. Una vez más, utilizando la arquitectura estándar de RIO NI y LabVIEW, se pueden crear rápidamente prototipos de sistemas embebidos de forma modular y flexible con CompactRIO y descargarlos rápidamente sobre el nuevo hardware embebido a nivel de tarjeta y de bajo costo NI Single-Board RIO. Puesto que es posible reutilizar el mismo código de LabVIEW desde la creación del prototipo hasta la implantación, se puede acortar el tiempo de lanzamiento e incrementar la fiabilidad del dispositivo embebido y de la máquina. Los nuevos productos NI Single-Board RIO ofrecen las siguientes características:

- Adquisición y control embebido en una sola tarjeta.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

- Programación gráfica mediante LabVIEW y herramientas de drivers para la conexión de programas y aplicaciones con el fin de acelerar el desarrollo.
- Procesador incorporado de tiempo real para un funcionamiento fiable e independiente y el procesamiento de señal.
- Chip incorporado de FPGA para la personalización del procesamiento y de la sincronización de las E/S.
- E/S analógicas y digitales incorporadas.
- Sistemas de bajo costo para el diseño de sistemas embebidos a nivel de tarjeta.

Cada dispositivo NI Single-Board RIO integra un procesador en tiempo real embebido, una FPGA de alto rendimiento y E/S analógicas y digitales en una sola tarjeta. Al igual que el resto del hardware RIO de NI, las E/S se conectan directamente a la FPGA, proporcionando una personalización a bajo nivel de la temporización y del procesamiento de las señales de las E/S. La FPGA está conectada al procesador embebido de tiempo real a través de un bus PCI de alta velocidad. LabVIEW contiene mecanismos de transferencia de datos para transmitir los datos de las E/S a la FPGA y de la FPGA al procesador embebido para el análisis en tiempo real, el post-procesado, el registro de datos o la comunicación con el ordenador host conectado en red.

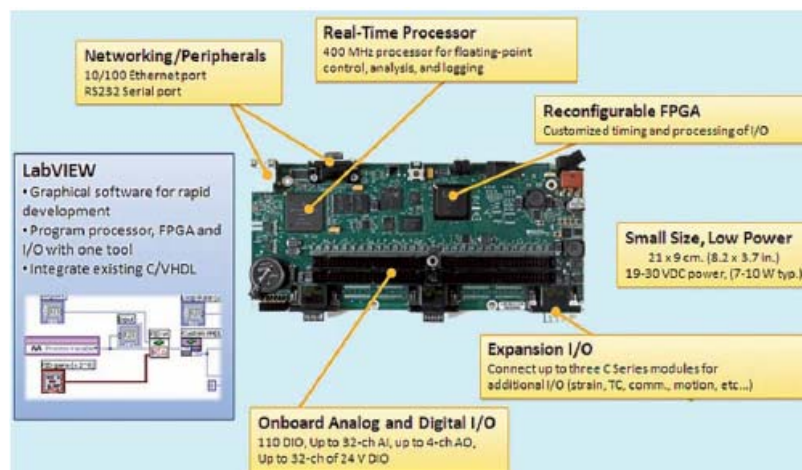


Figura 4: NI Single- Board RIO integra un procesador, una FPGA y E/S analógicas y digitales en una sola tarjeta que se puede programar con las herramientas gráficas de LabVIEW.

El coste estas tarjetas puede alcanzar unos 400 € por unidad, para unos pedidos a partir de 50 unidades.

### 3.5.2 Distribución más rápida con los sistemas embebidos con LabVIEW.

Con el entorno gráfico de desarrollo de LabVIEW se puede programar el procesador de tiempo real, la FPGA reconfigurable y las E/S de los sistemas

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

integrados RIO para las aplicaciones de control embebido, monitorización, procesamiento y registro utilizando el mismo proyecto de LabVIEW.

Con los módulos especializados de LabVIEW se puede programar el procesador de tiempo real (módulo LabVIEW Real-Time) y la FPGA (módulo LabVIEW FPGA).

LabVIEW tiene también un amplio conjunto de drivers para conectar entre si el software y las aplicaciones lo cual hace que la integración de todos los componentes del hardware del sistema embebido RIO (las E/S analógicas y digitales, la FPGA, el procesador, los periféricos y la memoria) sea un proceso fluido.

### **3.5.3 El módulo Labview Real-Time.**

El procesador embebido de tiempo real dentro de los sistemas está programado con LabVIEW Real-Time Module, que incluye bloques de función incorporados para el control, procesamiento, análisis, registro de datos y comunicaciones en coma flotante.

El módulo LabVIEW Real-Time incluye características tales como:

- Tecnología de variables compartidas de LabVIEW para la fácil creación de redes de sistemas distribuidos de tiempo real embebido.
- Software determinístico que permite una temporización con resolución del microsegundo.
- Más de 600 funciones de control avanzado y de procesamiento de señal en coma flotante.
- Integración del código C/C++ existente.
- Herramientas de replicación del sistema para descargar y duplicar rápidamente los sistemas existentes.

### **3.5.4 Módulo Labview FPGA.**

La FPGA reconfigurable dentro de los sistemas de hardware RIO se puede programar rápidamente con LabVIEW FPGA Module para realizar la personalización, el control de alta velocidad, la temporización de las E/S y el procesamiento de señales.

LabVIEW FPGA Module incorpora las siguientes características:

- IP a nivel de componentes (CLIP) Nodo para la fácil integración del código HDL existente.
- Asistente de FPGA para crear rápidamente la arquitectura de tiempo real y el código de la FPGA.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

- Asistente de proyectos basados en FPGA para que el comienzo sea más fácil.
- LabVIEW Statechart Module para implementar el control basado en FPGA, máquinas de estados y otras más.
- La simulación de FPGA facilita el desarrollo y la depuración.
- Bloques de funciones FPGA IP en coma fija.
  - Transformada rápida de Fourier (FFT)
  - PID multicanal.
  - Generadores de señal.
  - Filtro Notch.
  - Etc

### **3.5.5 Herramientas de drivers de conexión de programas y aplicaciones.**

Uno de los mayores retos en el diseño embebido es el esfuerzo necesario para crear, depurar y validar las pilas de software a nivel de driver para integrar todos los componentes de hardware del sistema embebido.

Tradicionalmente, este proceso de integración se le deja al usuario, lo que complica y alarga el proceso de diseño del sistema embebido.

Los drivers de conexión de programas y aplicaciones (middleware) de NI van más allá de los driver básicos que los ordenadores tradicionales de una sola tarjeta y otros proveedores de sistemas embebidos ofrecen para incrementar la productividad y el rendimiento y acortar el tiempo de lanzamiento del producto al mercado. Con cada dispositivo que soporta RIO se incluye el software del driver y el software adicional de los servicios configuración. Las herramientas incorporadas de drivers 'middleware' contienen las siguientes funcionalidades:

- Funciones incorporadas de interfaz entre las E/S analógicas, digitales, de movimiento y de comunicación y la FPGA.
- Funciones de transferencia para la comunicación de datos entre el procesador y la FPGA.
- Métodos de interfaz entre la FPGA y el procesador con la memoria.
- Funciones de interfaz entre el procesador y los periféricos (RS232, Ethernet).
- Drivers multi-hilo para obtener alto rendimiento.

### **3.5.6 Transición desde la creación del prototipo a la implantación.**

La plataforma RIO al disponer de múltiples factores de forma ofrece diversas opciones de creación de prototipos e implantación. El sistema integrado CompactRIO y NI Single-Board RIO proporciona las mejores características para los sistemas integrados de elevado volumen. Al igual que otros productos a nivel de tarjeta, el hardware embebido NI Single-Board RIO requiere que se garantice que el diseño se ajusta a los requisitos de conformidad con el estándar, como por ejemplo la compatibilidad electromagnética (EMC) y que se permita una adecuada disipación del calor. Los sistemas empaquetados y disponibles comercialmente CompactRIO proporcionan estas certificaciones.

### **3.5.7 Alto rendimiento con la programación con el Simple Ciclo While Loop (SCLP).**

La mayor parte de los conceptos relacionados con la programación de FPGAs de alto rendimiento en LabVIEW implican el uso eficaz de la estructura Simple Ciclo While Loop (o SCT). El SCTL es una estructura FPGA LabVIEW clave que reduce el uso de recursos y permite un mayor rendimiento y de control de temporización más precisa. El SCTL ofrece un paradigma de programación diferente que se aproxime más al comportamiento de los circuitos FPGA y proporciona un mayor control sobre la aplicación real del código de LabVIEW en la FPGA.

- El SCTL versión estándar para Código Labview FPGA

Cuando se programa con el Módulo LabVIEW FPGA, el contenido de tu diagrama se traslada al hardware, por lo que, en general, cada nodo en el diagrama tiene una representación equivalente en la FPGA como un componente del circuito. Cuando se coloca el código fuera del SCTL, LabVIEW controla la ejecución de los componentes de hardware como los datos fluyen a través de ellos. Como resultado, se necesita circuitería adicional para asegurarse de que estos componentes se ejecutan sólo cuando tienen datos válidos en todas sus entradas. Este modelo de ejecución se refiere a al estructura de flujo de datos.

El flujo de datos estructurado sigue un modelo tradicional de la ejecución del programa, donde una función debe tener todos sus parámetros de entrada antes de ejecutarse. Al ejecutar el código en una CPU, la CPU ha fijado, un propósito general, los circuitos secuencialmente consume códigos y datos. Sin embargo, a diferencia de una CPU, el código sobre una FPGA se convierte en un circuito altamente especializado y en paralelo, y los flujos de datos a través de él como señales eléctricas.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

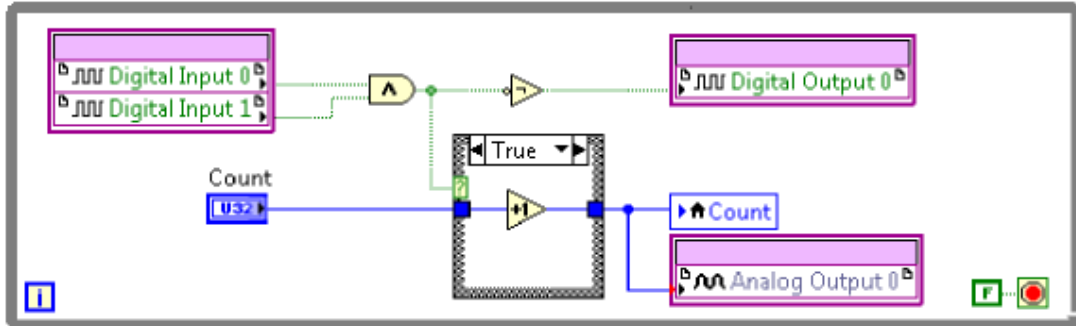


Figura 5: Código de Labview FPGA que sintetiza los circuitos y ejecuta secuencialmente cuando existen dependencia de datos.

Cuando el código se compila en un While Loop, LabVIEW inserta registros de hardware o elementos de almacenamiento pequeños, a los datos de reloj de una función a la siguiente, haciendo cumplir de esta manera la naturaleza del flujo de datos estructurada de LabVIEW.

Estos registros de elementos son añadidos al control de ejecución de la función y asegurado de datos en cada ciclo de reloj del flujo de datos de un nodo a otro. Cada nodo puede tener uno o más ciclos a ejecutar. En las aplicaciones de LabVIEW FPGA, el código situado fuera de la SCTL puede exhibir varios comportamientos temporales en una compilación a través de versiones diferentes del Módulo LabVIEW FPGA.

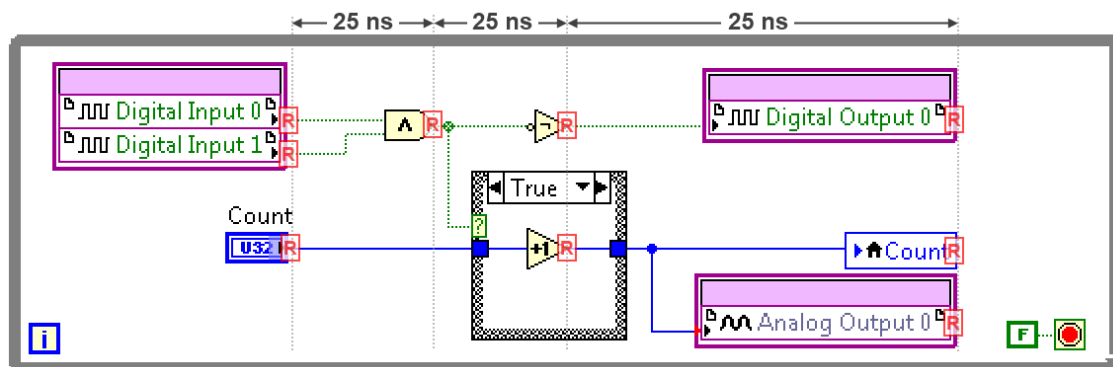


Figura 6: El código LabVIEW FPGA utiliza registros en cada función para enganchar los datos en cada ciclo del reloj de nivel superior, normalmente de 40 MHz (período de 25 ns). Aquí, los registros de hardware están representados por cajas etiquetadas "R."

El preservar el flujo de datos estructurado significa que, para las cadenas de nodos con dependencias de datos, sólo un subconjunto de los nodos está ejecutando activamente en un momento dado, mientras que el resto de la circuitería esta a la espera de datos.



La ejecución secuencial puede parecer un uso ineficiente de los circuitos FPGA dedicado, pero es suficiente para muchas aplicaciones. El modelo de flujo de datos estructurados proporciona una forma para que los desarrolladores de LabVIEW puedan escribir el código de LabVIEW que se asemeja a lo que crean en el escritorio, y permite a los desarrolladores ejecutar ese código en la FPGA sin preocuparse de los detalles de implementación. Como resultado, los desarrolladores de LabVIEW FPGA gozan de un gran rendimiento, la determinación, y la integración de Entradas y salidas.

– Comprensión de la estructura SCTL

La comprensión de la funcionalidad de la SCTL y el código colocado en el interior es la clave para la creación de aplicaciones de LabVIEW FPGA de alto rendimiento. El SCTL es una estructura única para aplicaciones de LabVIEW FPGA. Aunque la estructura SCTL y el código colocado en el interior se ven casi idéntica a la de LabVIEW Timed Loop y código interno relacionado, el código colocado dentro de la SCTL realiza de una manera muy diferente, ya que está garantizado para ejecutar dentro de un ciclo de reloj de un reloj FPGA específica. El SCTL representa los siguientes cinco conceptos clave:

○ Estructura

Estas estructuras tienen un significado específico en LabVIEW. Proporcionan alcance y puntos de transición definidos para los datos a medida que fluye a través de sus fronteras. El SCTL es una estructura y sigue el modelo de flujo de datos estructurado con respecto a otras estructuras en el diagrama. En concreto, el SCTL no puede empezar a ejecutarse hasta que todas sus entradas, o cables que entran a través de túneles o registros de desplazamiento, se hayan recibido los datos. Del mismo modo, la SCTL no produce salidas a través de sus túneles hasta que el código interior ha completado su ejecución.

○ Loop

El SCTL no es sólo una estructura que define cómo se ejecuta el código, pero también es un bucle de programa. Se ejecuta repetidamente su contenido y que obedece a While Loop que ejecuta al menos una vez, con una condición de parada que se puede establecer en tiempo de ejecución.

○ El reloj

Cada SCTL debe tener un reloj. El reloj define la frecuencia que se utiliza para conducir el circuito sintetizado a partir de los contenidos SCTL. Debido a que todos los circuitos dentro de las acciones SCTL tienen el mismo reloj, también se le llama un dominio de reloj.

La capacidad de definir el reloj SCTL es el mecanismo principal para el uso de los relojes múltiples dentro de un diseño. Diferentes SCTLs pueden utilizar

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

diferentes relojes que definen múltiples dominios de reloj. La velocidad del reloj y la fuente se deben determinar en tiempo de compilación, por lo que no se pueden cambiar una vez que el diseño se ejecuta.

- o Máxima iteración Latente

El SCTL no sólo especifica el reloj utilizado para conducir el código que encierra, pero también requiere que todo el código cerrado ejecutar dentro de un ciclo de reloj. 5.

- o Diferente Paradigma de Ejecución

El paradigma de la ejecución de la SCTL difiere de la versión estándar en LabVIEW porque todo el código cerrado se ejecuta en un ciclo de reloj. Para cumplir el requisito de iteración latente, LabVIEW FPGA compilación elimina circuitería de control de flujo utilizado para ejecutar el flujo de datos estructurado.

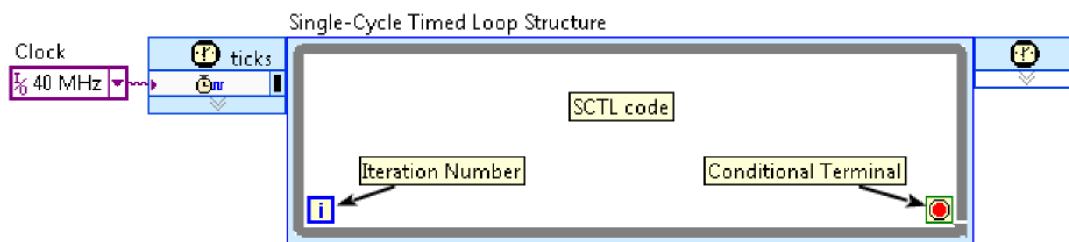


Figura 7: El SCTL es una estructura de bucle que especifica el reloj para ser utilizado por el código y asume una latencia máxima iteración de un ciclo de reloj.

La eliminación de los circuitos de control de flujo no sólo reduce el uso de recursos FPGA pero, lo más importante, permite que el diagrama se comporte como un circuito concurrente, donde los datos se representa mediante señales eléctricas que fluyen, sin restricciones, en todo el SCTL. Las señales se transforman por la lógica y están enganchados en ciertos puntos en el diagrama, tales como I / O, elementos de almacenamiento, y controles e indicadores.

Debido a que las señales están enclavadas en estos puntos específicos dentro del diagrama, deben establecerse un valor determinado antes de la llegada de la próxima transición de reloj, que marca el comienzo de una nueva iteración. Este comportamiento de enclavamiento, en combinación con el modelo de bucle del bucle, crea un circuito síncrono que se sintetiza por la cadena de herramientas de compilación FPGA.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

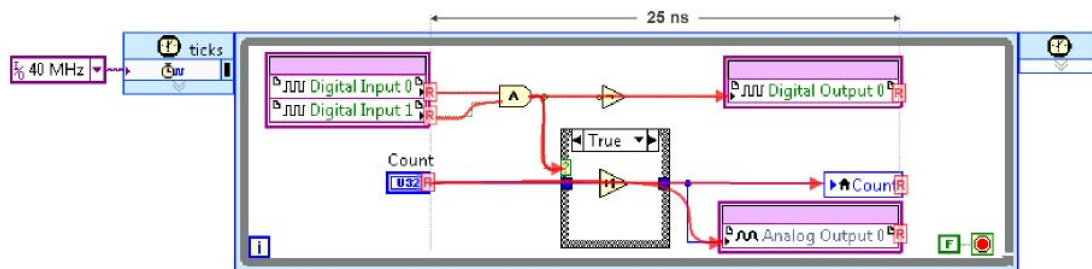


Figura 8: El SCTL representa un circuito síncrono donde las señales fluyen a través y están enclavadas sólo en puntos específicos como el de E / S, los controles y los indicadores y otras construcciones. El SCTL especifica el reloj utilizado para conducir sus contenidos y las señales deben propagarse de entre registros en menos de un ciclo de reloj, que es de 25 ns para este ejemplo

El término "síncrono" en circuitos síncronos difiere de su significado en los lenguajes de programación estándar cuando se refiere a los mecanismos de la función de las llamadas. Circuitos síncronos son simplemente circuitos accionados por un reloj, mientras que un síncronos llamada se bloquea la función de la ejecución de la persona que llama hasta que la función devuelve un resultado.

#### – Beneficios de la STCL

La latencia de la iteración de un ciclo que la SCTL especifica que se transmite de padres a la compilación cadena de herramientas Xilinx, junto con el código VHDL generado en tiempo de compilación. El compilador de Xilinx sintetiza el código SCTL y trata el requerimiento de un ciclo como una restricción de circuito. Si la compilación tiene éxito, el diseño generado está garantizado para ejecutar y liquidar todas sus señales dentro de un ciclo de reloj. Este comportamiento difiere de la del LabVIEW Timed Loop en una CPU, donde el período de ejecución no está necesariamente garantizada y deberá ser verificada en tiempo de ejecución.

La restricción de la latencia de una iteración es una parte importante del paradigma SCTL porque difiere de código LabVIEW FPGA estándar, que a menudo tiene más de un ciclo para ejecutar una sección de código. Debido a esta garantía sincronización, puede utilizar el SCTL como un mecanismo para especificar el tiempo entre los acontecimientos y la tasa de ejecución general, de la siguiente manera:

- El tiempo máximo entre dos eventos contenidos dentro de la misma iteración del SCTL está limitada a un período de bucle único. Esto es útil si usted está tratando de especificar una latencia máxima entre esos eventos.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

- Puede determinar, limitar, ni garantizar un número mínimo de ciclos entre los acontecimientos en iteraciones sucesivas. Esto es útil para medir y controlar eventos en aplicaciones tales como los protocolos de comunicación digitales.
  - Si usted sabe que la tasa de bucle está garantizada por el compilador, se puede prescribir el rendimiento. El bucle, en combinación con el número de muestras procesadas por iteración, proporciona una medida de rendimiento.
- Restricciones del SCTL

A pesar de que el código SCTL puede ser similar a otro código de LabVIEW, el paradigma de la programación y ejecución requiere un ajuste a su forma de pensar y cómo utiliza diferentes construcciones para lograr un mejor rendimiento.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

#### 4. Comparativa con PLCs tradicionales.

La plataforma CompactRIO cuenta con más de 60 módulos de entradas y salidas, incluyendo decenas de módulos con acondicionamiento de señal incluido (filtros, amplificación, compensación CJC, etc.). Usted puede encontrar un módulo de la serie C para prácticamente cualquier medición:

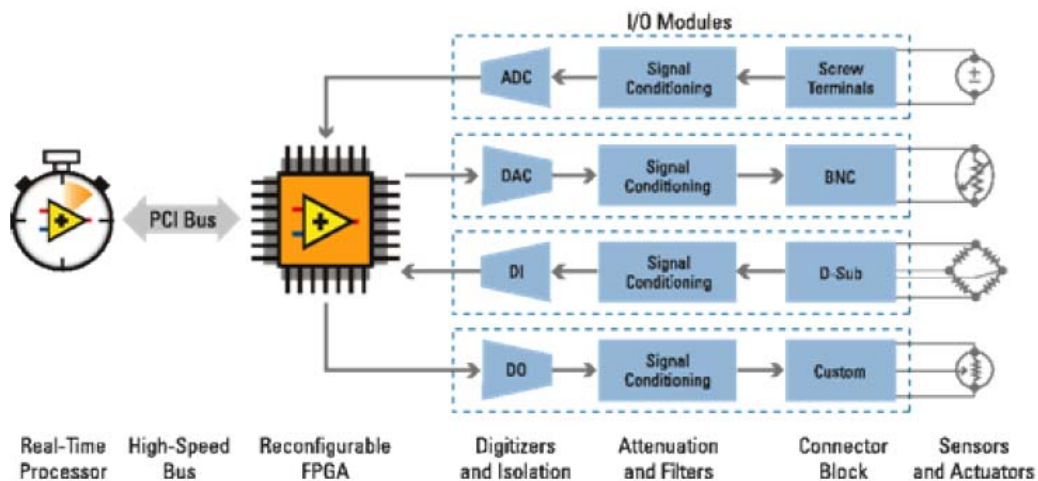


Figura 9: Estructura procesador Compact RIO

- Temperatura: termopares o RTDs
- Entradas y salidas digitales TTL o de nivel industrial (24V)
- Voltaje hasta 300V, corriente hasta 5A
- Vibración y sonido
- Mediciones de puente (galgas, celdas de carga, sensores de presión)
- Protocolos específicos de comunicación digital
- Etc.

Los módulos de entradas analógicas de la serie C pueden tener desde 12 hasta 24 bits de resolución y adquirir a una velocidad de hasta 1 MS/s/canal.

Otras características que posee:

- Entradas y salidas analógicas de alta velocidad y resolución.
- Capacidad de procesamiento: hasta Intel I7 dual-core a 1.33 Ghz.
- LabVIEW, el entorno de desarrollo gráfico para PACs de NI, maneja de una manera natural técnicas de programación más avanzadas, basadas en computación multinúcleo y de hilos múltiples. Esto le ayuda a programar sistemas de control que no serían posibles con una sintaxis más restringida como 61131.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

- Los componentes CompactRIO son fabricados usando aluminio, no plástico y el chasis es un componente sólido colado a diferencia de los sistemas estilo expansión que tienen múltiples uniones. El resultado es un sistema de control sólido que puede ser montado de manera segura en su máquina, panel o cubierta.
- Robustez absoluta, gracias al FPGA de la arquitectura RIO, que permite ejecutar las tareas más críticas directamente en hardware reconfigurable, eliminando posibles fallas de software.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

## 5. Equipo empleado.

El equipo empleado en los ensayos y prácticas es un myRIO, hardware diseñado para estudiantes, el cual dispone de unas herramientas de diseño amplias para investigación por un módico precio.

El funcionamiento es igual que los dispositivos RIO, CompactRIO, y las tarjetas NI Single RIO.

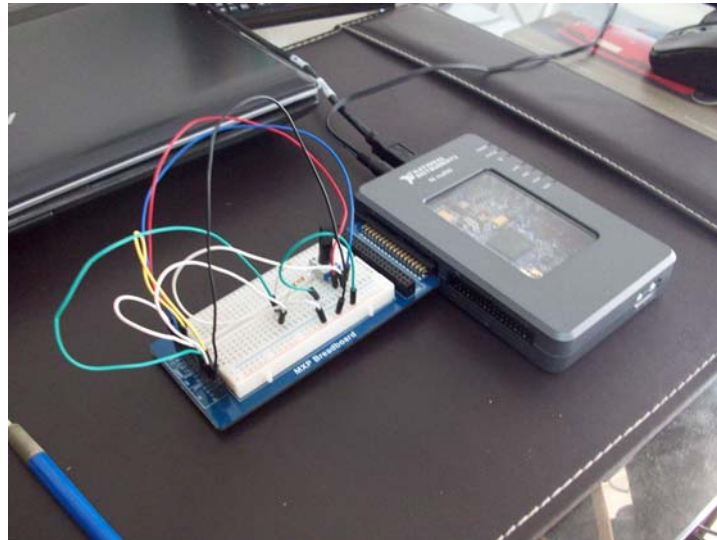


Figura 10: Equipo empleado en los ensayos. myRIO.

### 5.1 Descripción.

Para el desarrollo del proyecto se ha empleado el dispositivo de NI myRIO-1900, que es un dispositivo portátil y reconfigurable, principalmente de uso estudiantil y que te permite diseñar sistemas de control, robótica y mecatrónicos.

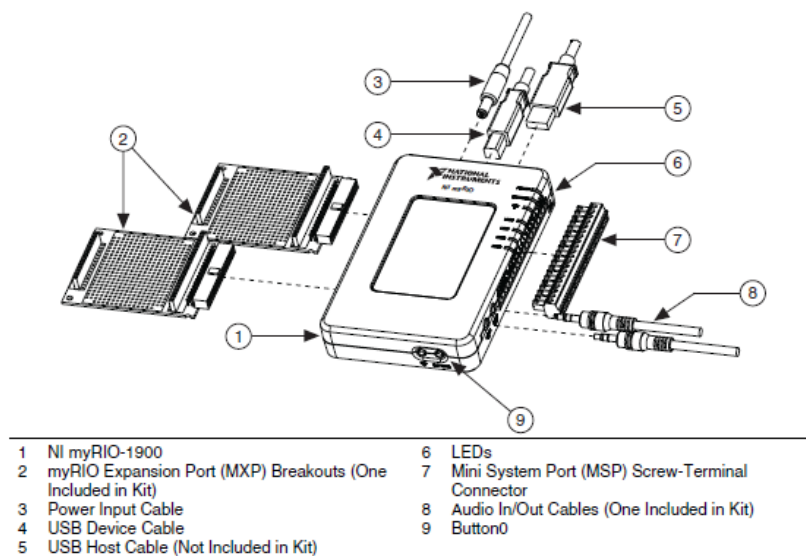


Figura 11: Partes del MyRIO

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

El procesador de que dispone es el tipo Xilinx Z-7010 con velocidad de 667 MHz y dos núcleos.

De memoria dispone de 256 MB no volátil, una memoria DDR3 de 512 MB, la frecuencia del reloj es de 533 MHz, y el ancho del bus de datos es de 16 bits.

El tipo de FPGA es la de la Xilinx Z-7010 y permite además conexión mediante Wifi o puerto USB serie.

Respecto a la velocidad de las entradas analógicas tienen una resolución de 12 bits y un porcentaje de muestras de 500 kS/s.

De forma genérica el hardware del dispositivo dispone de un procesador para el Real Time, una FPGA necesaria para el proyecto, y múltiples entradas/salidas analógicas y digitales como podemos observar en la figura siguiente:

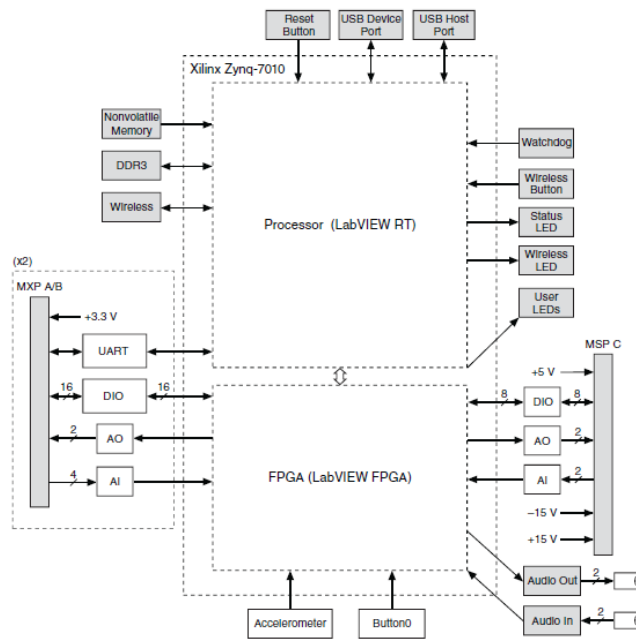
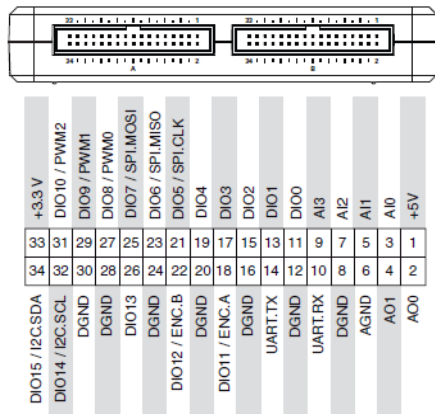


Figura 12: Estructura procesador MyRIO

Para las entradas de señal analógica, emplearemos los conectores A/B que tienen las siguientes características:



Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.



Signal Name	Reference	Direction	Description
+5V	DGND	Output	+5 V power output.
AI <0..3>	AGND	Input	0-5 V, referenced, single-ended analog input channels. Refer to the <i>Analog Input Channels</i> section for more information.
AO <0..1>	AGND	Output	0-5 V referenced, single-ended analog output. Refer to the <i>Analog Output Channels</i> section for more information.
AGND	N/A	N/A	Reference for analog input and output.
+3.3V	DGND	Output	+3.3 V power output.
DIO <0..15>	DGND	Input or Output	General-purpose digital lines with 3.3 V output, 3.3 V/5 V-compatible input. Refer to the <i>DIO Lines</i> section for more information.
UART.RX	DGND	Input	UART receive input. UART lines are electrically identical to DIO lines.
UART.TX	DGND	Output	UART transmit output. UART lines are electrically identical to DIO lines.
DGND	N/A	N/A	Reference for digital signals, +5 V, and +3.3 V.

Figura 13: Señales entrada y salida MyRIO

Las entradas analógicas son en los conectores MXP A y B, son multiplexadas a un convertidor analógico digital (ADC) para las muestras de todos los canales.

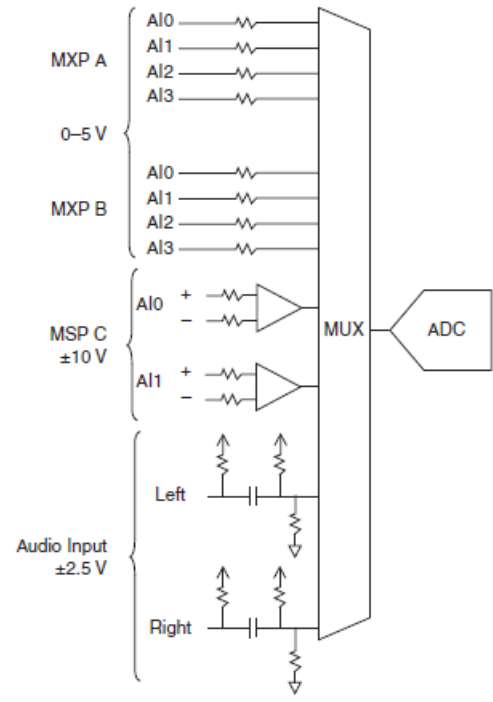


Figura 14: Composición del MyRIO multiplexados entradas/salidas analógicas.

En nuestro caso, las entradas a emplear miden entre 0-5 V

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

## 6. Metodología.

### 6.1 Configuraciones iniciales.

#### 6.1.1 Configuración WIFI sobre MyRIO.

Para poder conectar nuestro dispositivo en la red, vamos a configurar inicialmente el Wifi de este, para poder estar en la red.

Lo podemos llevar a cabo desde dos aplicaciones.

- NI Web-based Configuration & Monitoring
- Measurement & Automation Explorer (MAX)

Nosotros vamos a hacer desde MAX:

- Seleccionamos el dispositivo y con botón derecho seleccionamos Configuración web.

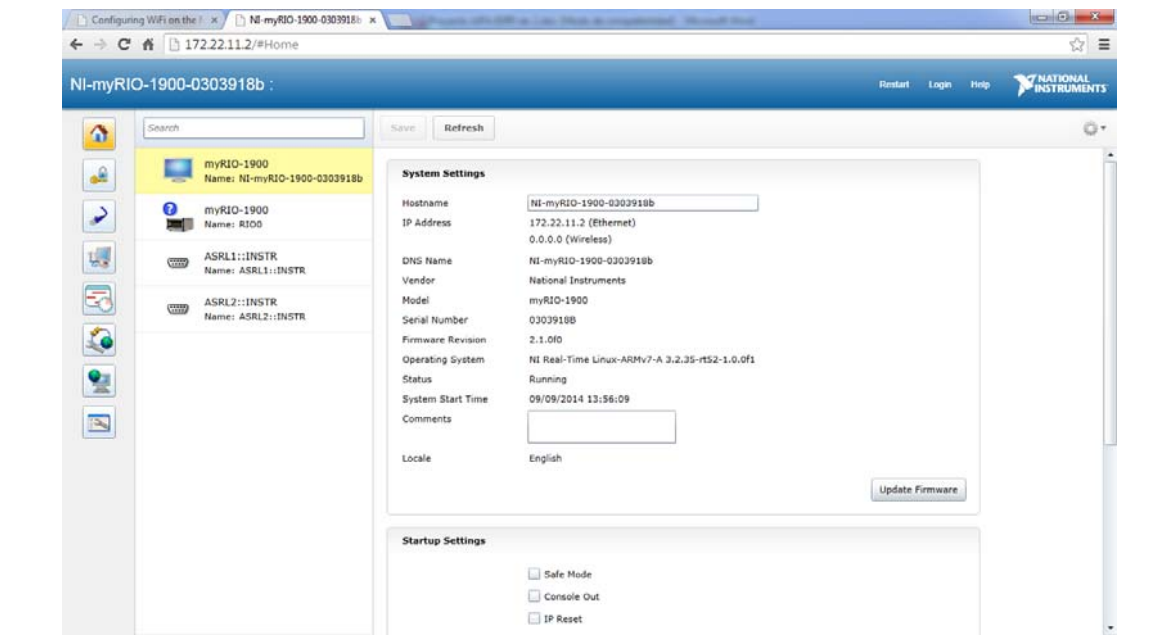



Figura 15: Esquema Configuración WEB del MyRIO.

- Nos vamos a configuración de red pulsando  y configuramos las características de configure WiFi settings in the Wireless Adapter wlan0.
- Seleccionamos Select Connect to wireless network in Wireless Mode.
- In Country, introducimos el país donde está localizado el equipo. Debemos seleccionar antes el país que el canal de red wireless.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

- Seleccionamos la red de la que dispongamos, posiblemente sea necesario una contraseña para acceder.
- Seleccionamos DHCP or Link Local in Configure IPv4 Address, y especificamos que NI myRIO adquiriera una dirección IP automáticamente.
- Guardamos la configuración WIFI y conectamos el NI myRIO a la red wireless que seleccionamos.
- Por último para verificar que está conectado NI myRIO a la red y adquirir la dirección IP, tu deberías usar la IP cuando despliegues VIS en el NI myRIO.

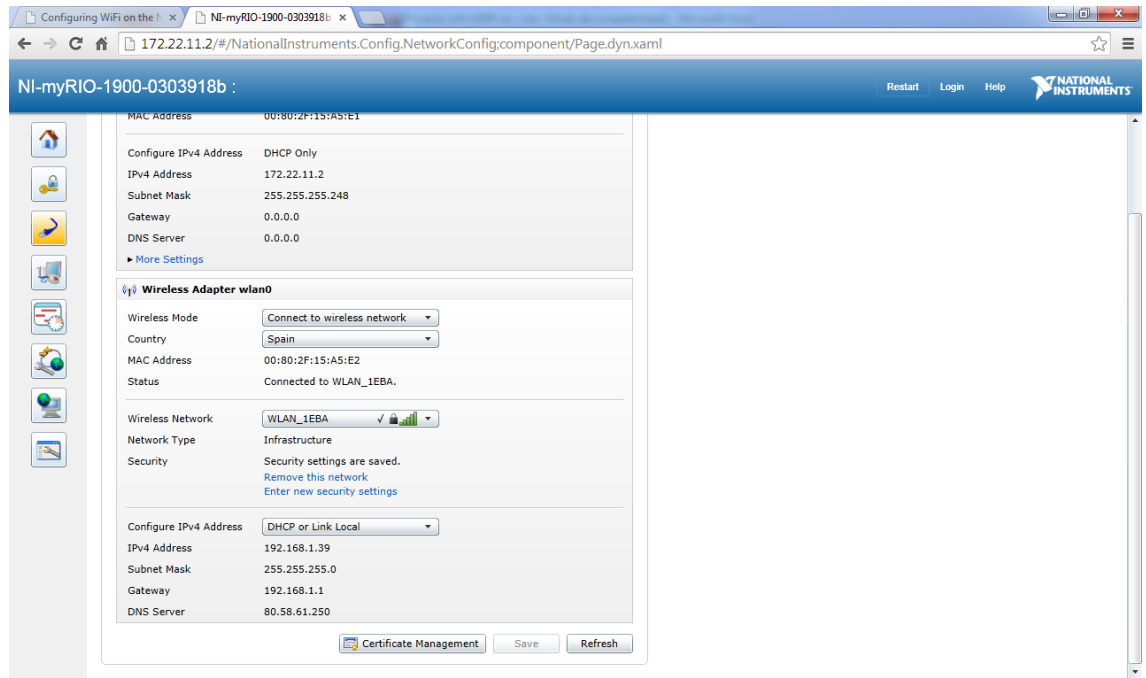


Figura 16: Esquema Configuración de las redes de conexión del MyRIO.

### 6.1.2 Configuración día y hora en MyRIO.

En la misma configuración web podemos hacerlo, debemos guardar los cambios.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

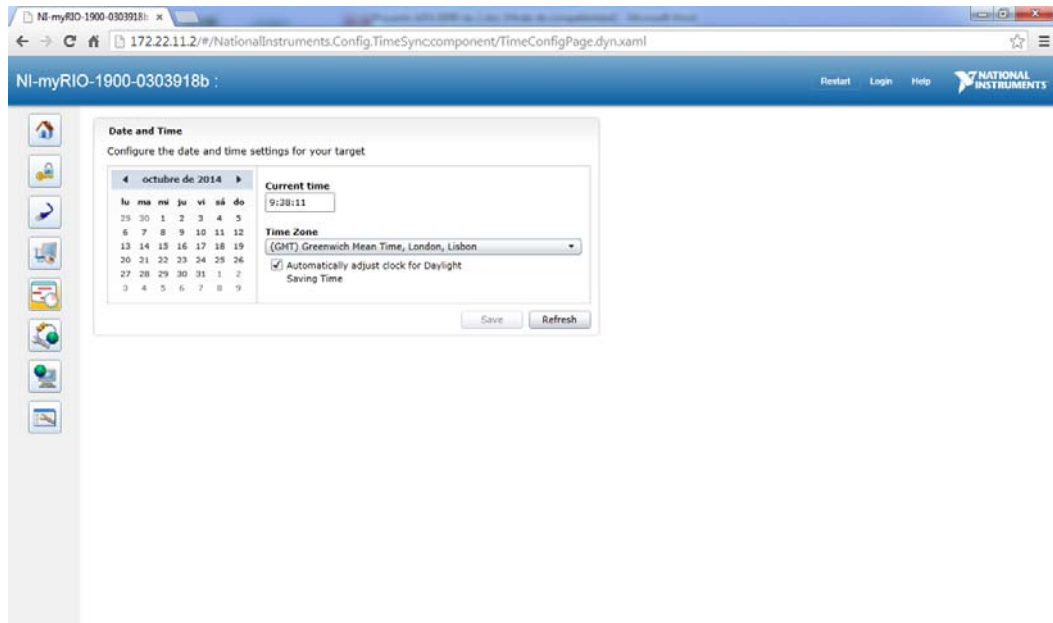


Figura 17: Esquema Configuración Horario del MyRIO.

## 6.2 Simulación de variables a medir.

Para simular las 4 señales a medir vamos a emplear el propio dispositivo que dispone de señal de 5 y 3,3 V de salida, y conectaremos un potenciómetro y unas resistencias de poco valor (10 , 20 y 6,8 K $\Omega$ )

Para la ejecución de los ensayos hemos usado una simulación de tensiones empleando el propio MyRIO y una de sus tarjetas MXP, ya que tiene una salida a 5 V.

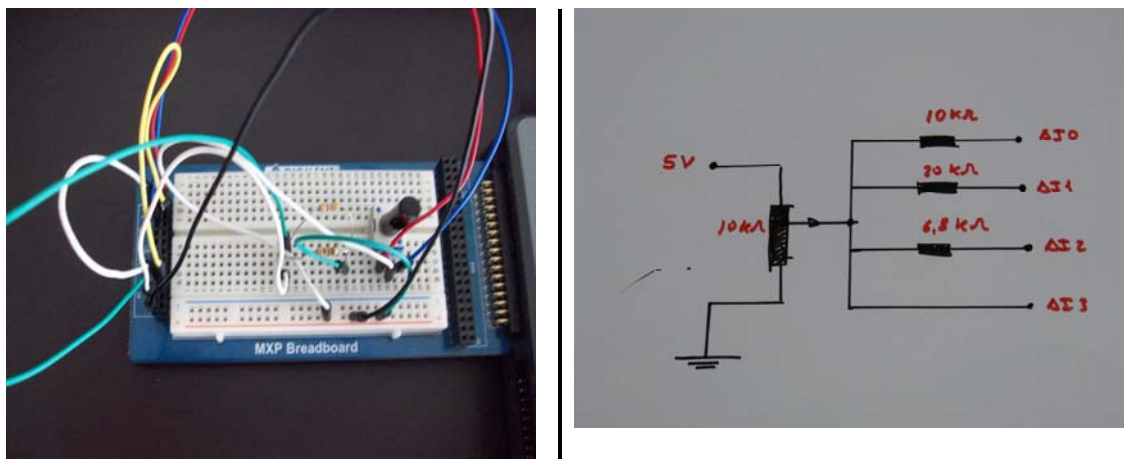


Figura 18: Esquema Montaje simulación señales

Hemos usado la salida de 5 V como fuente de tensión, y mediante un potenciómetro hemos regulado esa tensión de salida, y colocado una resistencia a cada entrada para que no coincidan entre ellas.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

### 6.3 Configuración dispositivo.

Inicialmente necesito añadir el dispositivo (hardware) que vamos a emplear en el diseño de nuestro proyecto a software de NI llamado MAX (Measurement & Automation Explorer). El dispositivo que vamos a emplear es el MyRIO.

Vamos a Remote System y desde allí añadimos un nuevo dispositivo llamado NI-myRIO 1900- 0303918b, con una IP Address 172.22.11.2.

Una vez que tenemos Remote System el dispositivo, debemos añadir el software necesario para desarrollar el proyecto.

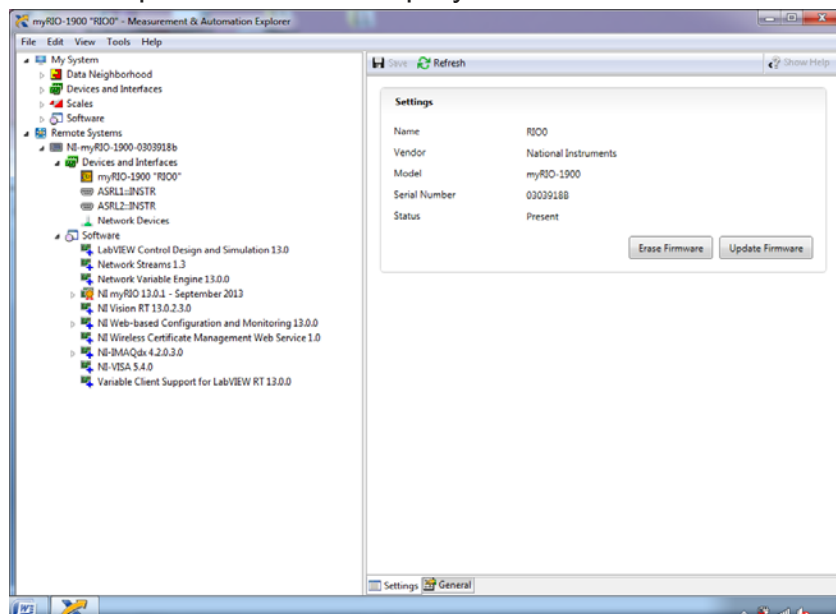


Figura 19: Configurar el Hardware MyRIO en MAX

Ahora debemos abrir Labview y crear un nuevo proyecto con myRIO.

Seleccionamos myRIO Custom FPGA Project.

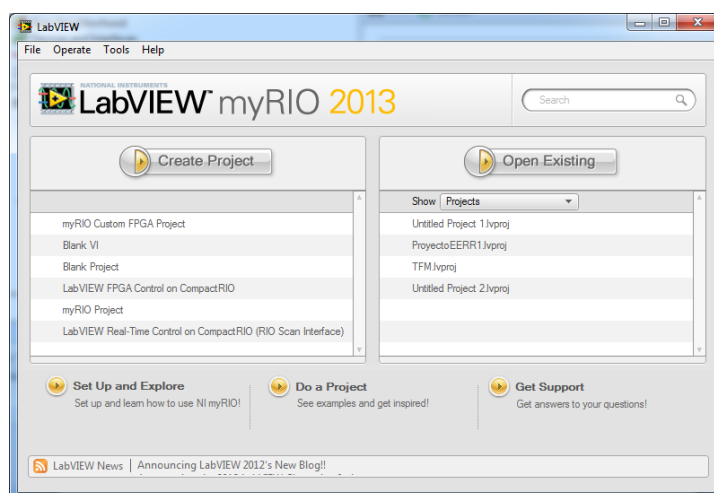


Figura 20: Entrada a Labview 2013.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

Para trabajar inicialmente en la programación vamos a seleccionar plugged into USB.

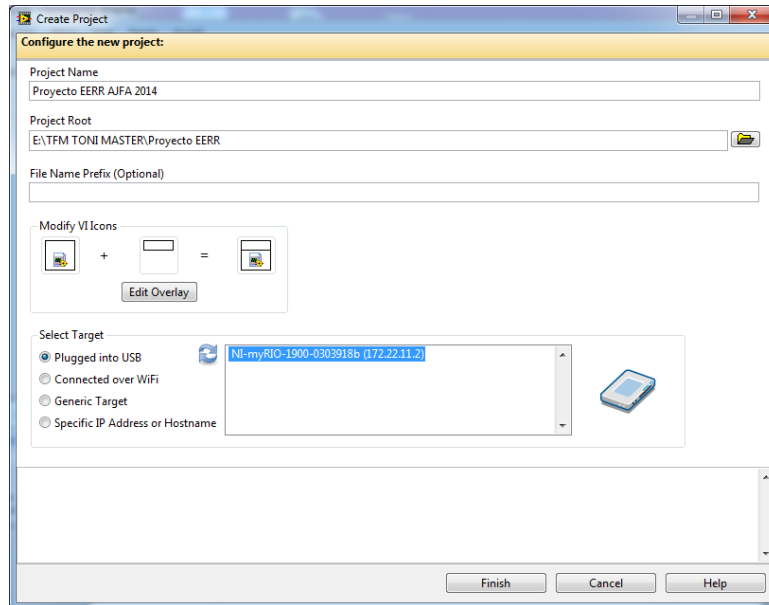


Figura 21: Esquema de conexión a MyRIO mediante USB

Ahora se nos abre Labview y podemos observar las partes de que se compone el MyRIO.

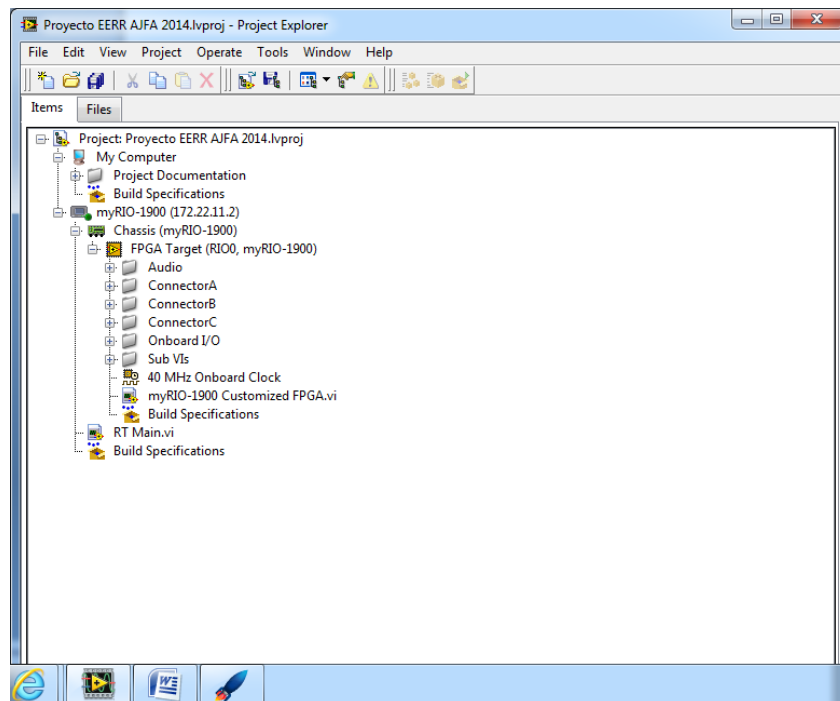


Figura 22: Esquema Carpetas Proyecto

Al seleccionar myRIO Custom FPGA Project nos añade todos los elementos del dispositivo, 3 conectores A, B y C, donde cada uno dispone de varias entradas y salidas, una entrada y salida de Audio, y una tarjeta Onboard que dispone de sensores de aceleración y LED. El dispositivo será explicado con más detalle en otro punto.

#### **6.4 Programación archivos VI de la tarjeta de adquisición de datos.**

Pretendemos programar la tarjeta de adquisición de datos del controlador para poder tomarlos bajo unas variables definidas como son la frecuencia de muestreo. Estos valores serán la tensión y la intensidad.

Para ello utilizaremos el propio controlador, mediante la programación de su FPGA y usando la herramienta que dispone FIFO (buffer de memoria). Esta herramienta permite adquirir una gran cantidad de valores a una frecuencia elevada con la posibilidad de procesar todos esos valores.

Por tanto se va a explicar la programación llevada a cabo en LabVIEW para conseguir nuestros objetivos.

Primeramente creamos un VI en la FPGA de nuestro myRIO, seleccionamos new y guardamos como tarjeta datos.

##### **6.4.1 Diagrama de bloques.**

- Vamos a crear un **bucle While Loop**, que nos repite una acción hasta que llegue una condición de parada que en nuestro caso será un control booleano de Stop.
- Creamos un indicador de número de ciclos (Loops).
- Creamos dentro del bucle While Loop, una estructura de secuencia, llamada **Flat Sequence Structure**. Esta estructura permite que los subdiagramas o Frame se ejecuten secuencialmente uno tras el otro.
  - Frame 1: Colocamos un control Count tipo ticks de reloj (5000) que se establece para temporizar el bucle uniéndolo a un Loop Timer, que temporiza el bucle para ejecutarse.
  - Frame 2: Como se necesita abrir los canales de medida y leer en modo continuo, desde la pantalla de Project Explorer arrastramos los canales que queremos medir, que en nuestro caso será los del Conector A: A/AI0, A/AI1, A/AI2 y A/AI3. Para poder visualizar los datos en tiempo real, colocamos un indicador de datos, indicándole que el tipo de datos sea tipo FXP. Le cambiamos el nombre de Intensidad, Tensión, Velocidad y Dirección (viento).

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

- Para visualizar los datos en tiempo real medidos, tenemos que hacer una corrección de los datos leídos en la FPGA por el hardware, y multiplicarlos por 1,221 mV.

## Converting Raw Data Values to Voltage

You can use the following equations to convert raw data values to volts:

$$V = \text{Raw Data Value} * \text{LSB Weight}$$

$$\text{LSB Weight} = \text{Nominal Range} \div 2^{\text{ADC Resolution}}$$

where *Raw Data Value* is the value returned by the FPGA I/O Node,

*LSB Weight* is the value in volts of the increment between data values,

*Nominal Range* is the absolute value in volts of the full, peak-to-peak nominal range of the channel,

and *ADC Resolution* is the resolution of the ADC in bits. (*ADC Resolution* = 12.)

- For AI and AO channels on the MXP connectors,

$$\text{LSB Weight} = 5 \text{ V} \div 2^{12} = 1.221 \text{ mV}$$

$$\text{Maximum reading} = 4095 * 1.221 \text{ mV} = 4.999 \text{ V}$$

- For AI and AO channels on the MSP connectors,

$$\text{LSB Weight} = 20 \text{ V} \div 2^{12} = 4.883 \text{ mV}$$

$$\text{Maximum Positive Reading} = +2047 * 4.883 \text{ mV} = 9.995 \text{ V}$$

$$\text{Maximum Negative Reading} = -2048 * 4.883 \text{ mV} = -10.000 \text{ V}$$

Figura 23: Esquema conversión voltaje de MyRIO

- Los datos FXP o número de coma fija se da en aplicaciones de FPGA colocaremos como longitud total del número 12 bits y para la parte entera 3 bits.

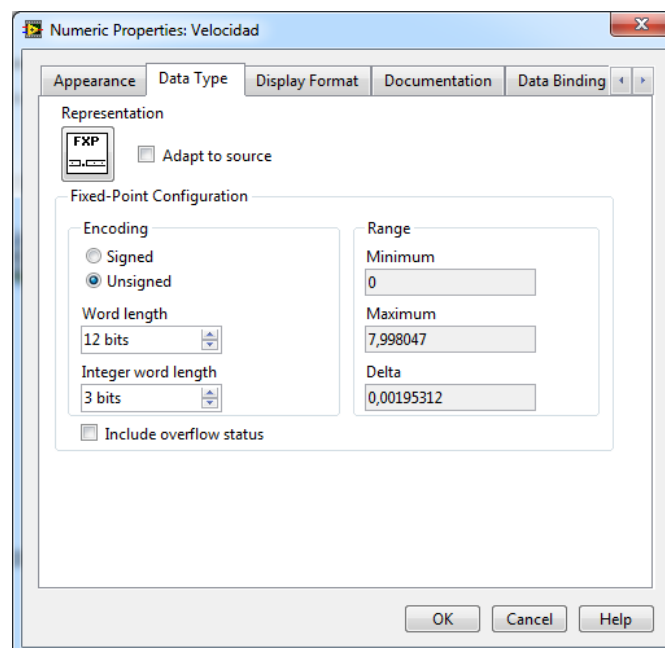


Figura 24: Propiedades del tipo de dato.



- Creamos un **FIFO**: Antes de crearlo necesitamos crear un array, elementos que son del mismo tipo desde una a muchas dimensiones y conteniendo hasta  $2^{31} - 1$  elementos en cada dimensión. Usamos la función Build Array, y unimos cada canal a medir y obtenemos un array de 1 D con los canales a medir.

La ventaja de emplear la función Build Array en la FPGA es que no requiere ciclos de reloj para ejecutarse, ya que no incluye un registro interno, y además esta función consume fuentes no lógicas sobre la FPGA porque es solamente una operación de cableado.

- Creamos un **bucle FOR** para fijar el número que queremos que se repita el bucle.

Volvemos a la creación del FIFO.

- **Continuación FIFO**: Sobre la FPGA del dispositivo pulsamos New → FIFO, y nos aparece la ventana con las propiedades de este.
  - Inicialmente debemos especificar que recurso de la FPGA se tiene y la transferencia de datos por la FIFO. Seleccionamos pues el modo Between the host computer and FPGA, Direct Memory Acces (DMA). Este es un método con base en la FIFO de transferencia de datos entre la FPGA y el ordenador. La DMA no implica al procesador del ordenador por lo que es el método disponible más rápido para transferir gran cantidad 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
  - Requested Number of element (número de elementos solicitados) : 1023
  - Data Type : U32
  - Dentro de Interface: Arbitratio for Write, seleccionamos Arbitrate if Multiple Requestor Only (seleccionado cuando usamos el write o read con bucle Timed Loop). El Number of elements per write → seleccionamos 1.

Ya esta creado el FIFO, ahora debemos arrastrarlo desde el proyecto hasta al diagrama de bloques.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

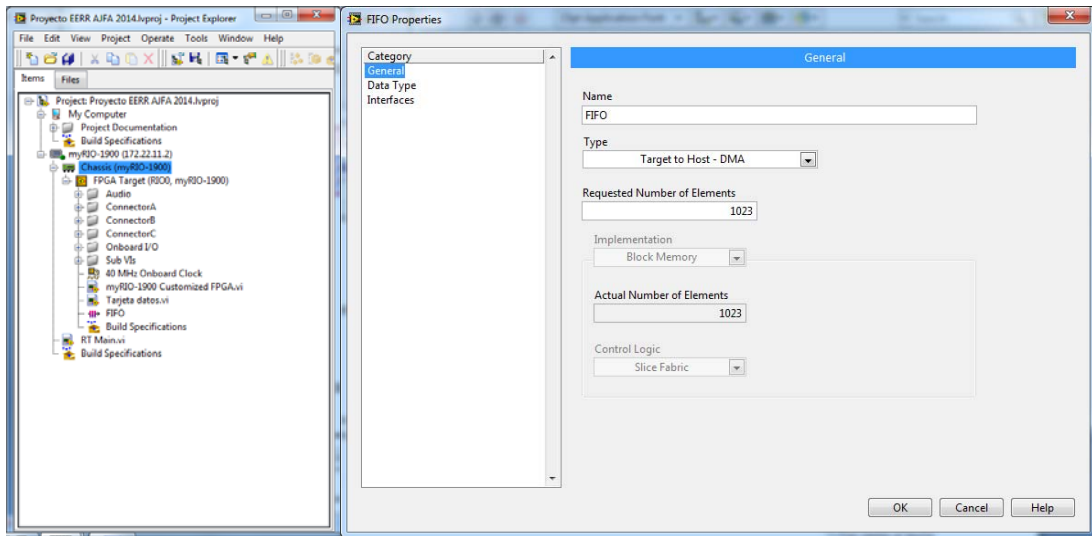


Figura 25: Características de la FIFO

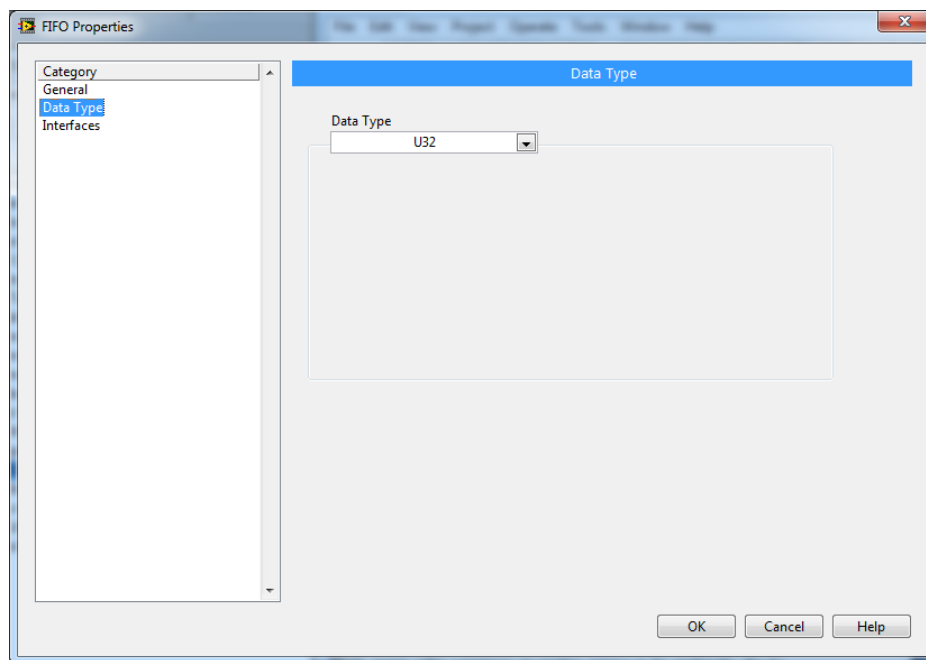


Figura 26: Datos de la FIFO.

Cuando se ha arrastrado dentro de nuestro bucle FOR por defecto se instala un FIFO tipo write. Mediante esta función lo que hacemos es escribir los valores dentro de la FIFO, para ello unimos nuestro array a la entrada de la FIFO element para que vaya tomando los valores. En el terminal Timeout se introduce el número de ticks de reloj que la función espera cuando la FIFO está llena, para un hueco disponible en la FIFO.

Si ponemos un valor 0, indicamos que no espere para sobre escribir en la FIFO cuando está llena. Si ponemos un valor -1, se configura indefinidamente hasta que se saque los valores de la FIFO.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

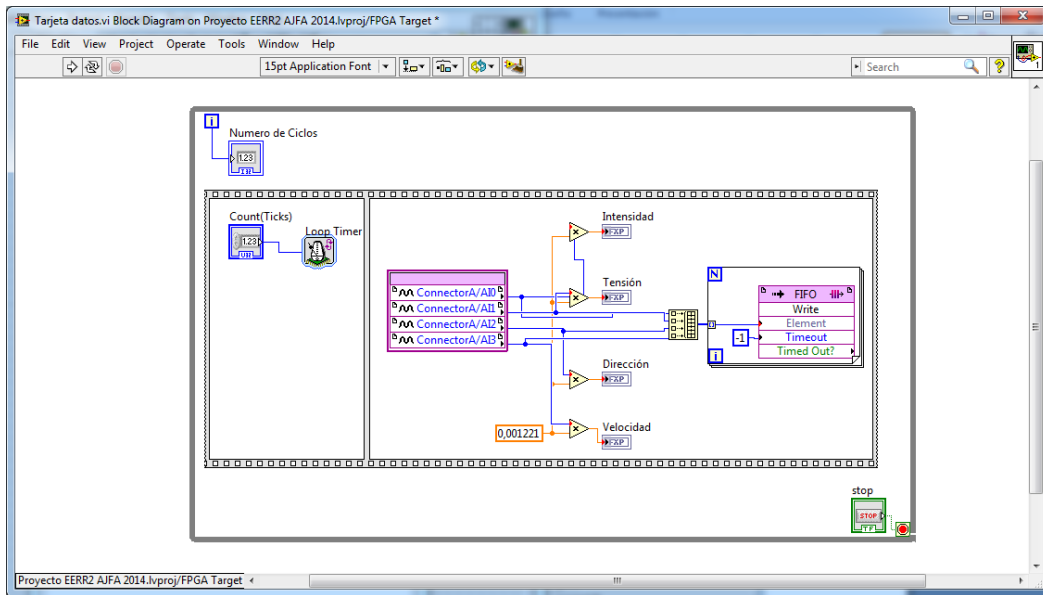


Figura 27: Diagrama de bloque VI de la FPGA.

#### 6.4.2 Panel de control.

En el panel de control nos aparecen todos los valores que se han programado en el diagrama de bloques:

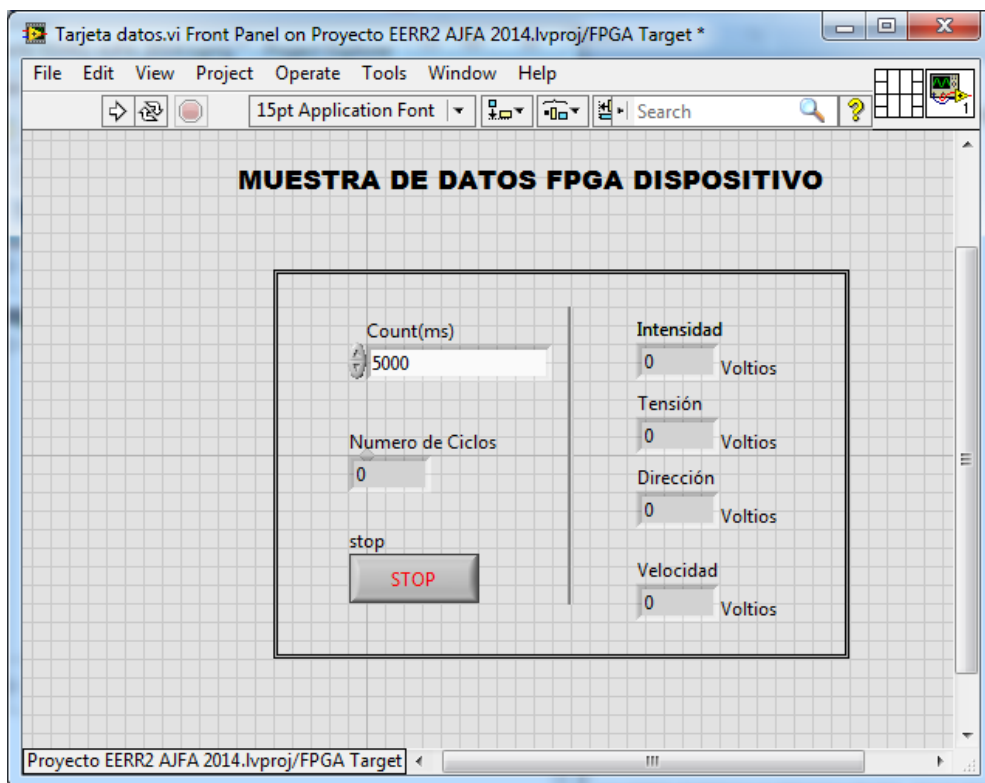


Figura 28: Panel de control tarjetas de datos de la FPGA.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

### 6.4.3 Compilación.

Una vez programado debemos compilar la programación en la FPGA del myRIO. En el Project Explorer con el botón derecho sobre el archivo tarjeta datos.vi seleccionamos Compile hasta que aparezca que ha compilado con éxito (suele tardar tiempo). Esta acción se hará de forma automática cada vez que modifiquemos algo en el archivo tarjeta datos.vi y se quiera grabar los datos. La duración aproximada es de 40 minutos.

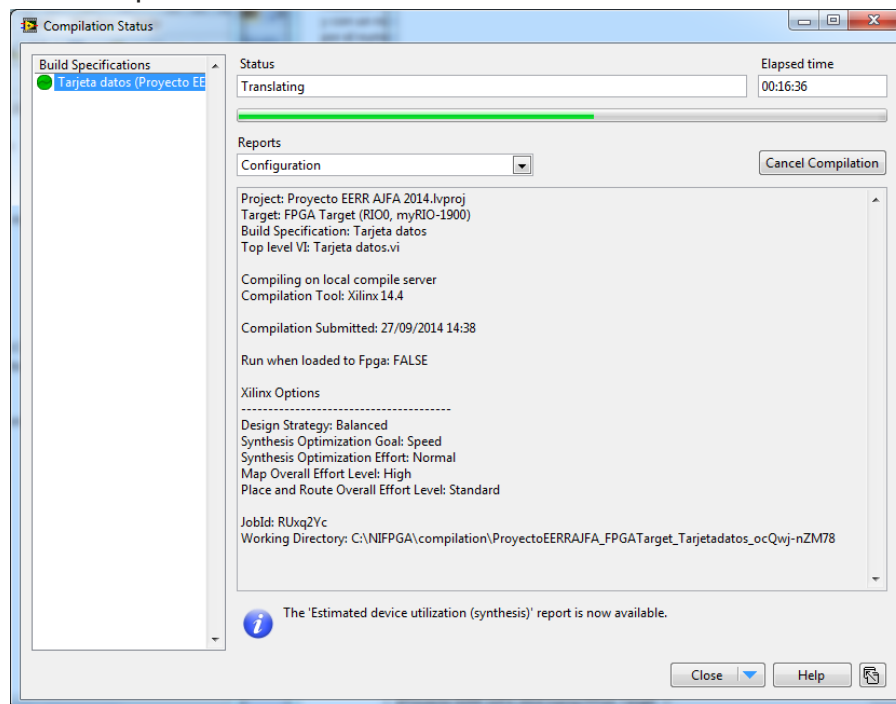


Figura 29: Compilación de la FPGA

## 6.5 Programación archivos VI del controlador CompactRIO.

Comenzamos creando un nuevo VI en la rama principal del myRIO, y lo vamos a llamar Medidas.vi. Sobre el dispositivo pulsamos botón derecho y seleccionamos new > VI. Lo llamaremos medidas.vi.

### 6.5.1 Diagrama de bloques.

- Lo primero es **abrir la FPGA**, mediante la función open FPGA VI reference (lo encontramos en FPGA interface)

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

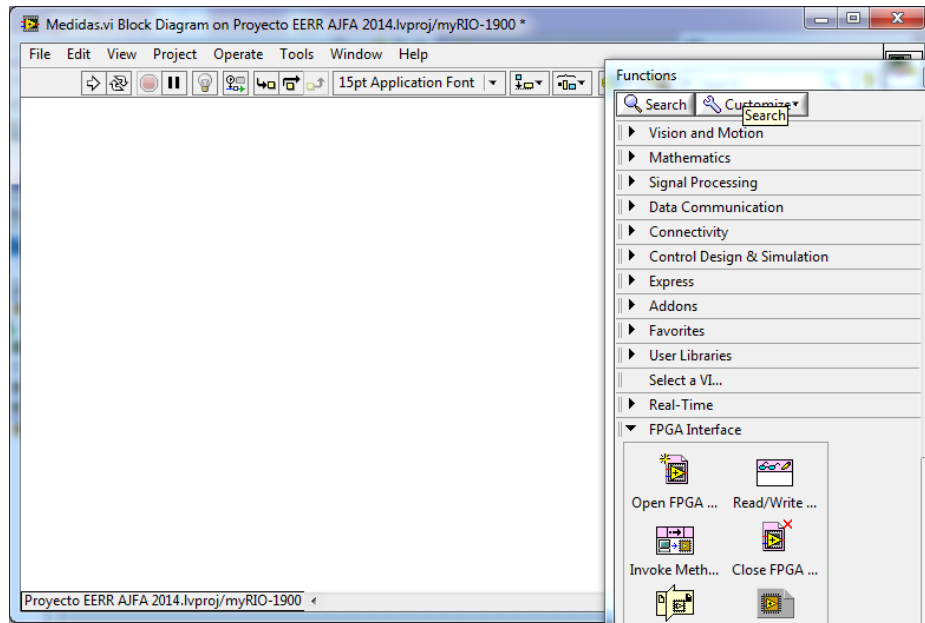


Figura 30: Insertar entrada FPGA

Posteriormente hay que configurarla, y seleccionar open > VI > e indicar la dirección de apertura, que en nuestro caso es tarjeta datos.vi

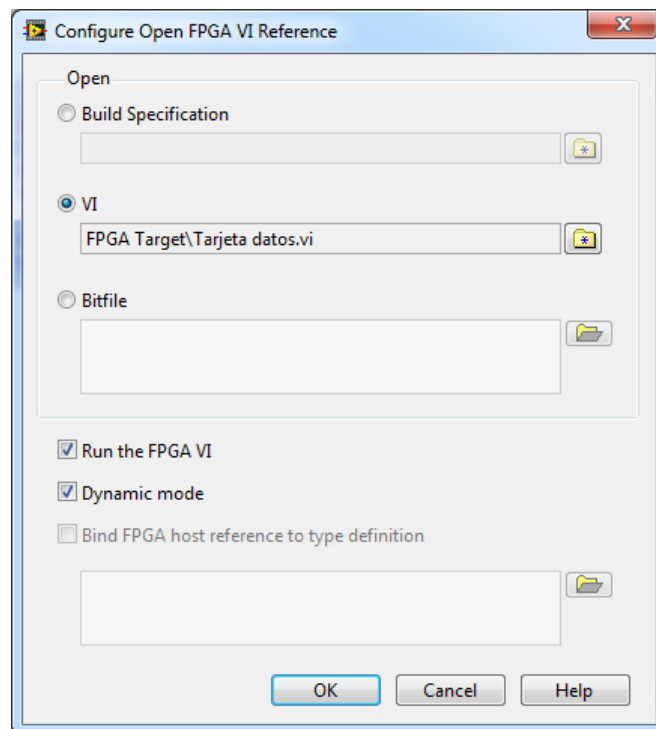


Figura 31: Configurar la FPGA que debe leer.

Inicialmente colocamos el FPGA VI reference y lo metemos dentro de una estructura Flat Sequence Structure para que se ejecute primero la Open FPGA.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

- Creación de un bucle Timed Loop

Usamos esta función por que nos permite ejecutar VIs con capacidades de tiempo multifrecuencia, y así ejecutar con un periodo específico varios VI con diferentes regímenes. En este caso se usa la FPGA VI, el bucle se ejecuta según el reloj interno de la FPGA.

El bucle Time Loop nos permite usar el reloj interno a 2 frecuencias: 1 kHz y 1 MHz. Con 1 kHz podemos hacer iteraciones cada 1 milisegundo, y con el reloj interno de 1 MHz podríamos hacer iteraciones cada 1 microsegundos. Para este último caso necesitaríamos un gran procesador (mínimo un Pentium III).

Inicialmente se va a establecer la prioridad de ejecución con un periodo de milisegundos a determinar por el usuario con una frecuencia de 1 KHz, con ello se pretende que se ejecute el bucle en el tiempo que quiera el usuario.

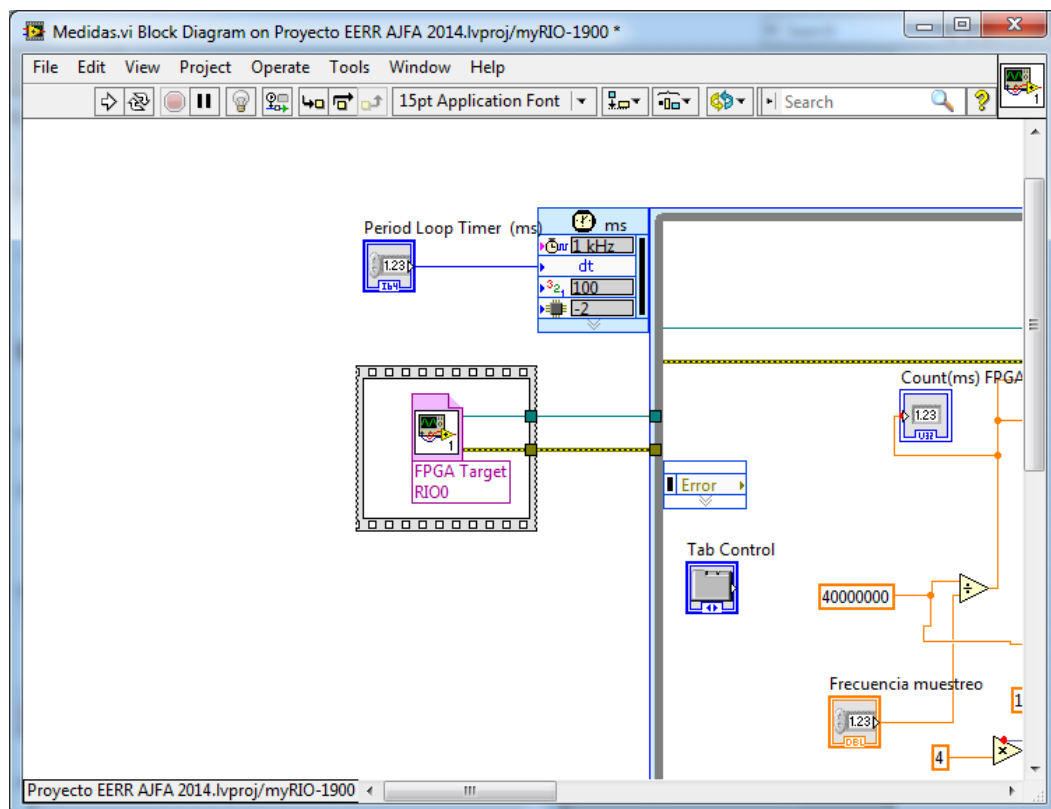


Figura 32: Creación del Loop Timer

La frecuencia del Loop Timer, también es determinada por el usuario (Periodo Loop Timer).

- Vamos a determinar la pauta de medición

Donde tomaremos una medida cada "X" segundos (a definir por el usuario) Para ello usaremos la función Quotient & Remainder Function, ya que requiere

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

ciclos de reloj y los registros en proporción al número de bits en x o y, cualquier tipo de datos es muy grande. Cada ciclo de reloj corresponde a un registro.

La x serán las iteraciones del bucle Timed Loop, pues bien lo que haremos será dividir x/y y cuando este valor sea cero, será nuestra condición de medida y lo uniremos al terminal de la estructura de casos Case Structure (opción True)

Además coloco un interruptor de No grabar, obligar a la estructura de casos a False y no guardar los datos, no leer.

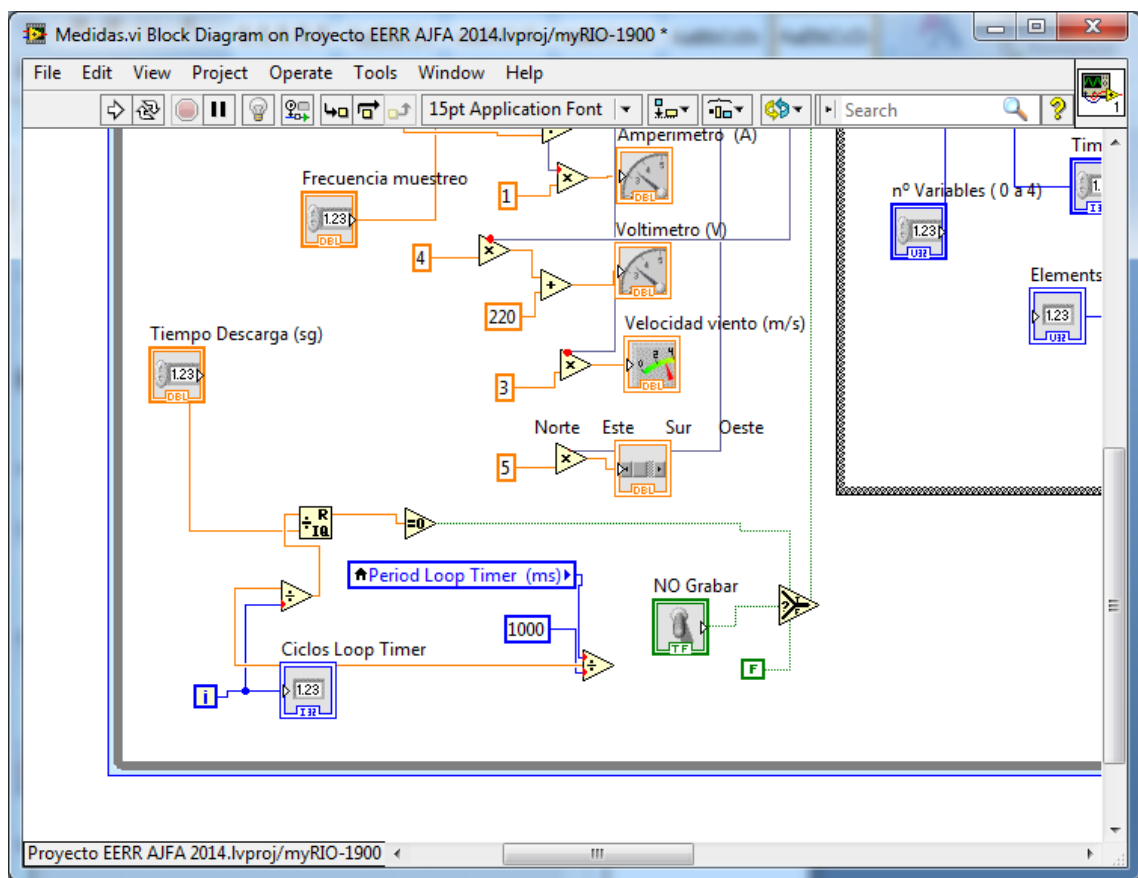


Figura 33: Configuración tiempo de descarga de datos de la FIFO.

- En la Estructura de casos, opción True, procedemos a insertar la función necesaria.

Inicialmente es necesario controlar la frecuencia de muestreo en relación con el reloj interno de la FPGA, ya que se utiliza la FIFO para aumentarla. Para ello emplearemos un control introduciendo la frecuencia de muestreo, por ejemplo 1 kHz y para sacar el tiempo de muestreo, divido la frecuencia de la FPGA entre la frecuencia de muestreo.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

- Por tanto, con 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**.

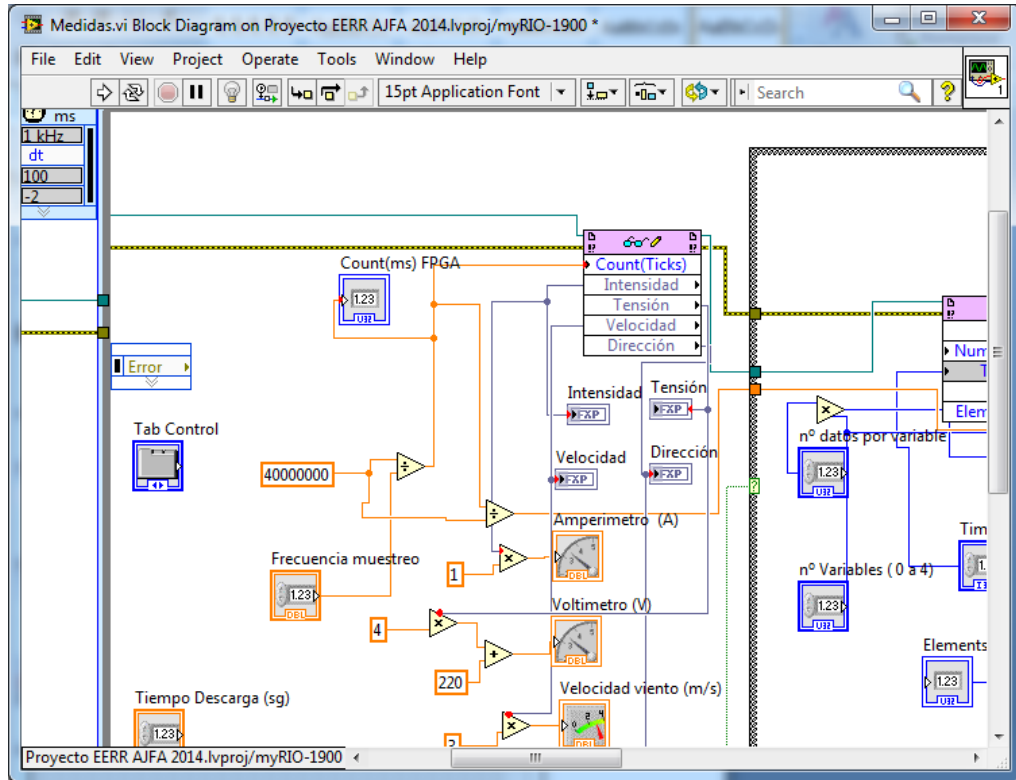


Figura 34: Configuración Control Read de la FIFO.

Además podemos observar que también se ha configurado la lectura de los datos en tiempo real mediante la opción Count Ticks.

Siempre que empleamos un FIFO, se necesita emplear dos funciones, write y read.

- El **FIFO Write** lo implementamos en el archivo tarjetas datos.vi y el **FIFO Read** dentro del controlador.

Este método devuelve valores cuando el número de elementos está disponible (la FIFO ha llenado su memoria) o cuando el periodo de espera termina.

Al FIFO Read le unimos los siguientes datos:

- Number of element: Serán de a definir por el usuario, por cada canal de medida, a la frecuencia de 1 KHz, siendo siempre un valor múltiplo del número de canales de medida.
- Timeout: Es el número de ms que el método Read espera a que estén las muestras, a definir por el usuario, colocaremos el valor 5000 por defecto.



Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

- Data: Devuelve los valores de los datos guardados en la memoria de la FIFO.
- Elements Remaining: Son los elementos que quedan esperando en la memoria hasta que se lean.

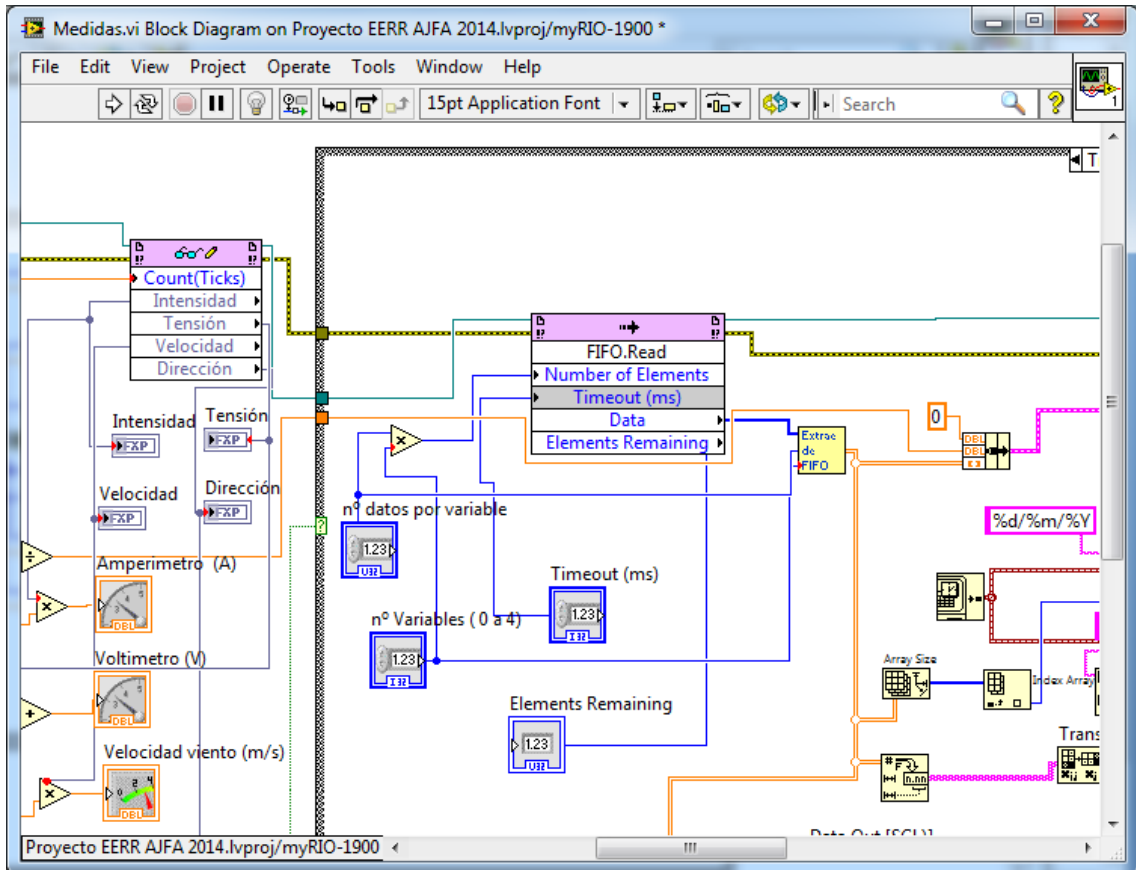


Figura 35: Con figuración de la FIFO Read.

Estos valores obtenidos, se conectan a un subdiagrama VI llamado extrae FIFO, que permitirá transformar el tipo de dato obtenido de U32 a SGL (punto flotante simple).

Con esta transformación podemos representar estos valores temporalmente en una gráfica.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

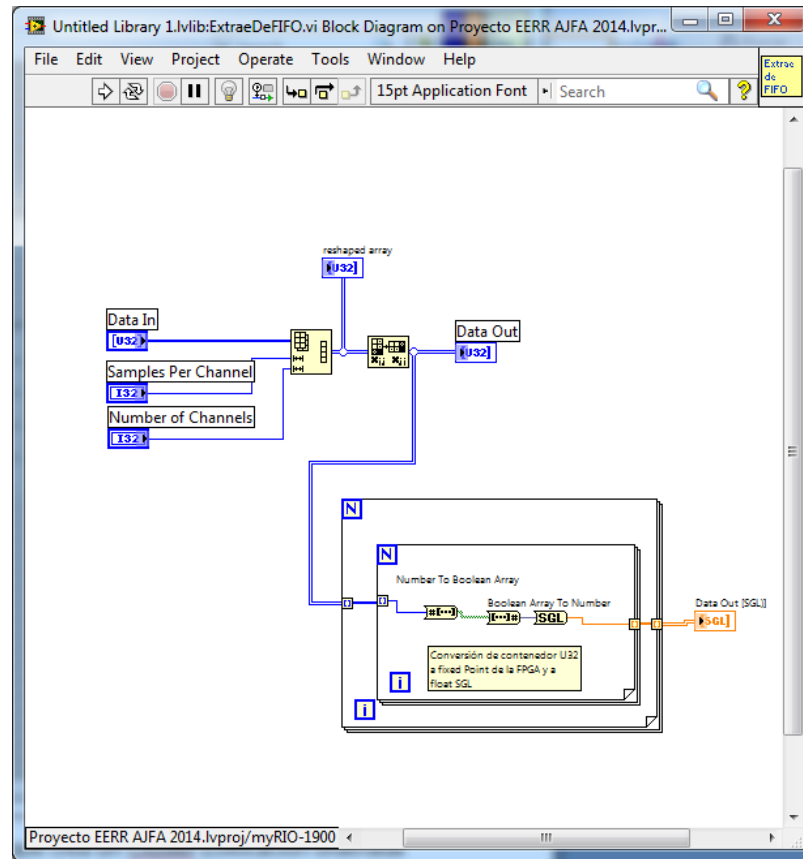


Figura 36: Diagrama de Bloque del Extraer de la FIFO.

Para ello usamos la función Bundle que crea un Cluster (colocación ordenada de uno o más elementos que pueden ser del mismo o diferentes 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 transformen 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 gráfico Waveform Graph. De esta manera se puede comprobar si los datos obtenidos son los que esperamos.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

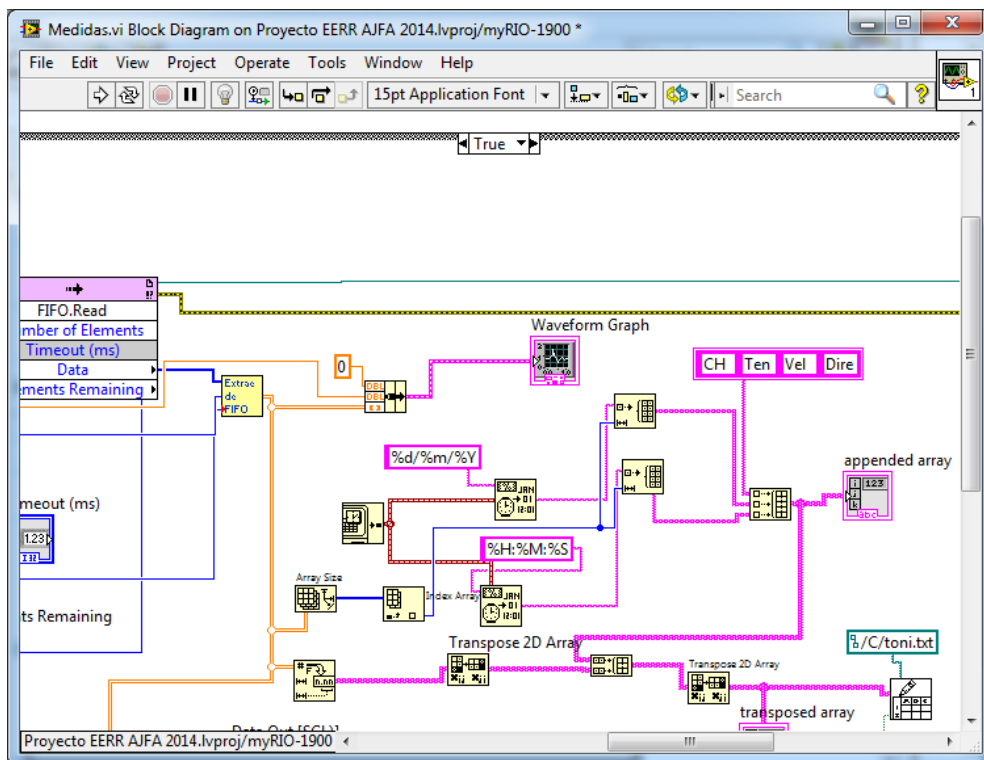


Figura 37: Configuración función Bundle para hacer una gráfico de los datos.

– Ahora procedemos a crear **un array** de fecha, hora y datos:

- 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 lectura de 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 de los mismos valores de fecha/hora en todos los elementos.

- 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.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

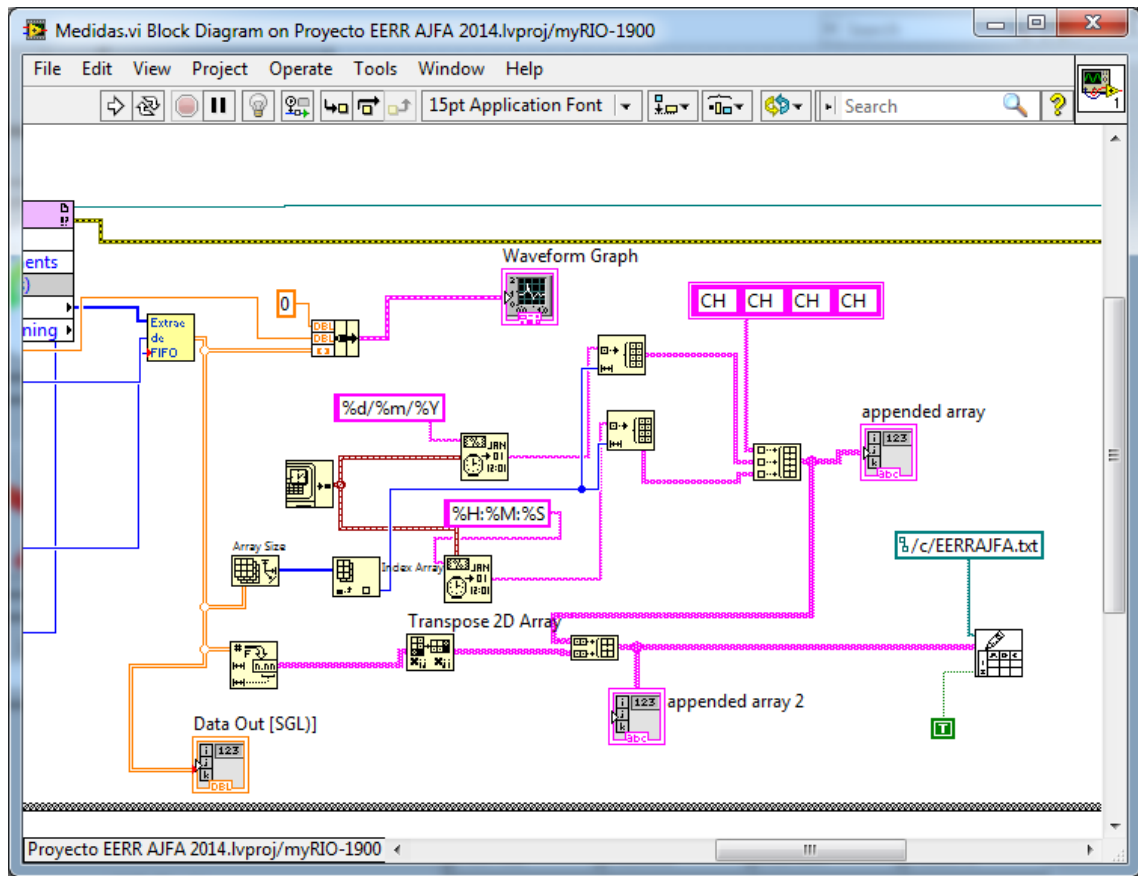


Figura 38: Creación de Array de fecha y hora.

Finalmente se guardan los valores en un .txt. Para ello el salvado de datos se hace 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 del dispositivo. En nuestro caso el archivo se llama EERRAJFA.txt, y se guarda en el disco C.

### 6.5.2 Panel frontal.

Finalmente podemos observar cómo se queda el panel frontal, que será nuestro SCADA, donde el usuario podrá interactuar y controlar el sistema de medidas

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

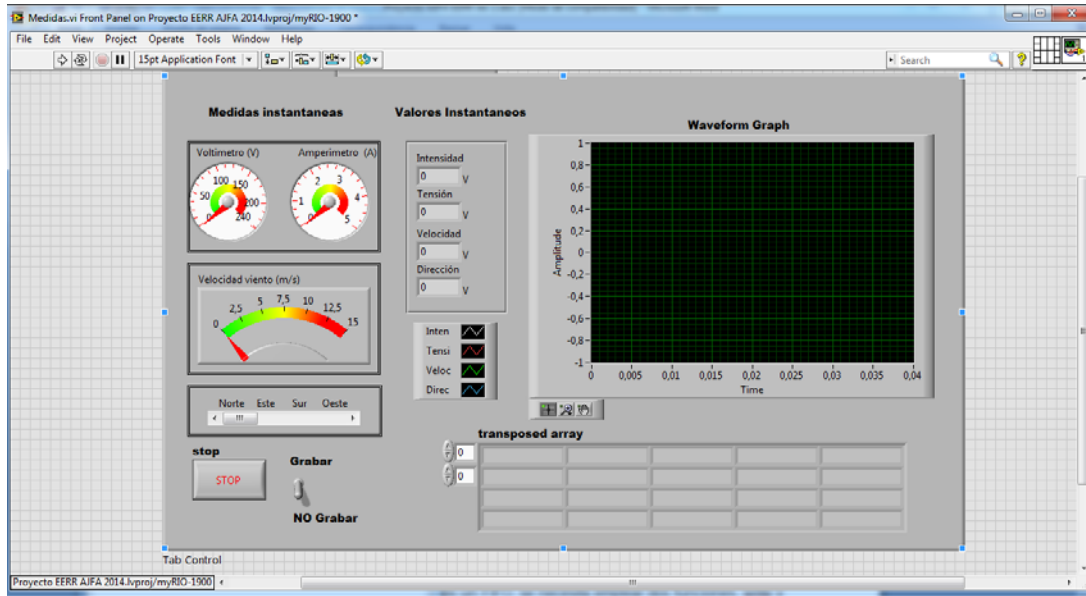


Figura 39: Panel de control principal del SCADA.

En él se puede observar la medidas tomadas por los sensores (0-5 V), los valores reales de medida, Intensidad, tensión, velocidad y dirección del viento.

La grafica de cómo varían los valores respecto al tiempo, los valores obtenidos, y un pulsador de parada del sistema, así como un interruptor de grabar o no grabar la FIFO.

En la obra pestaña del panel de control, tenemos los datos de configuración de la medida que queremos hacer.

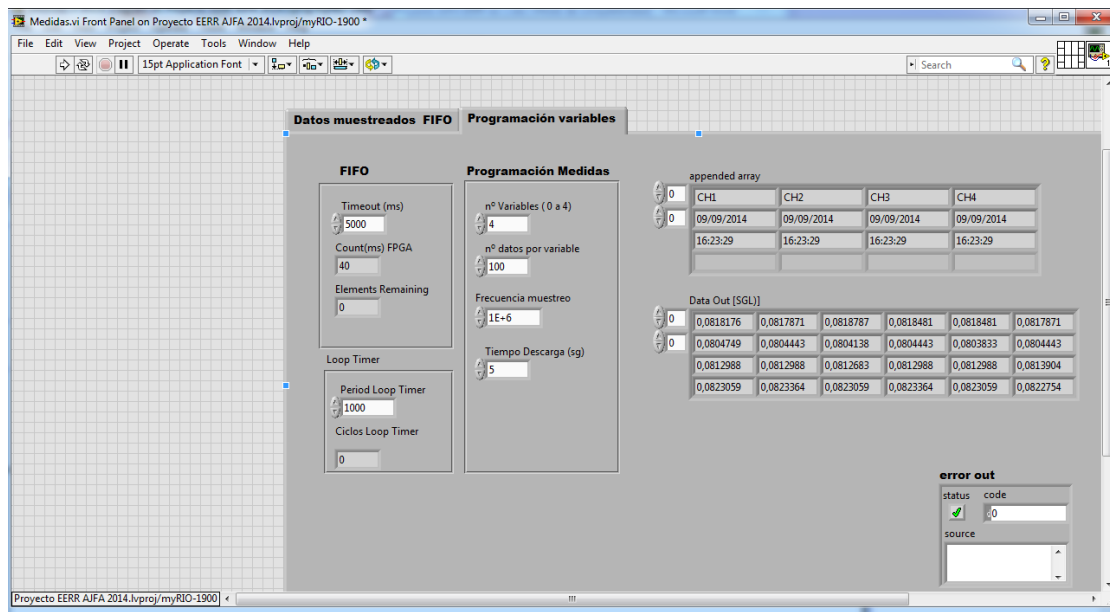


Figura 40: Panel de control secundario del SCADA.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

## 6.6 Comunicaciones.

### 6.6.1 Publicar el panel frontal en la web.

Con esta aplicación podremos actuar sobre el panel frontal desde un punto remoto.

- En primer lugar debemos habilitar la opción, en herramientas, opciones, web server y habilitar.

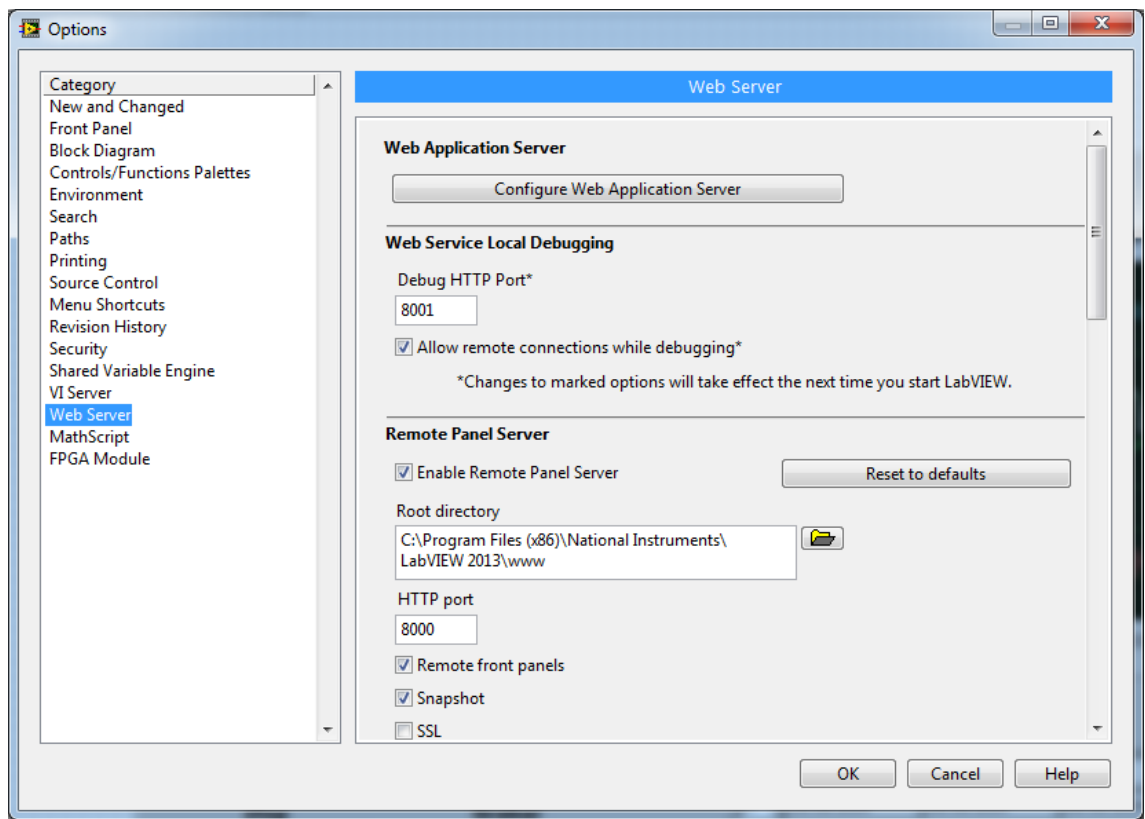


Figura 41: Configuración Web Server

- Posteriormente procedemos a publicar el panel frontal, en herramientas, web publishing web. Seleccionamos el VI que queremos publicar. Y tenemos 3 opciones para elegir en el modo de visión.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

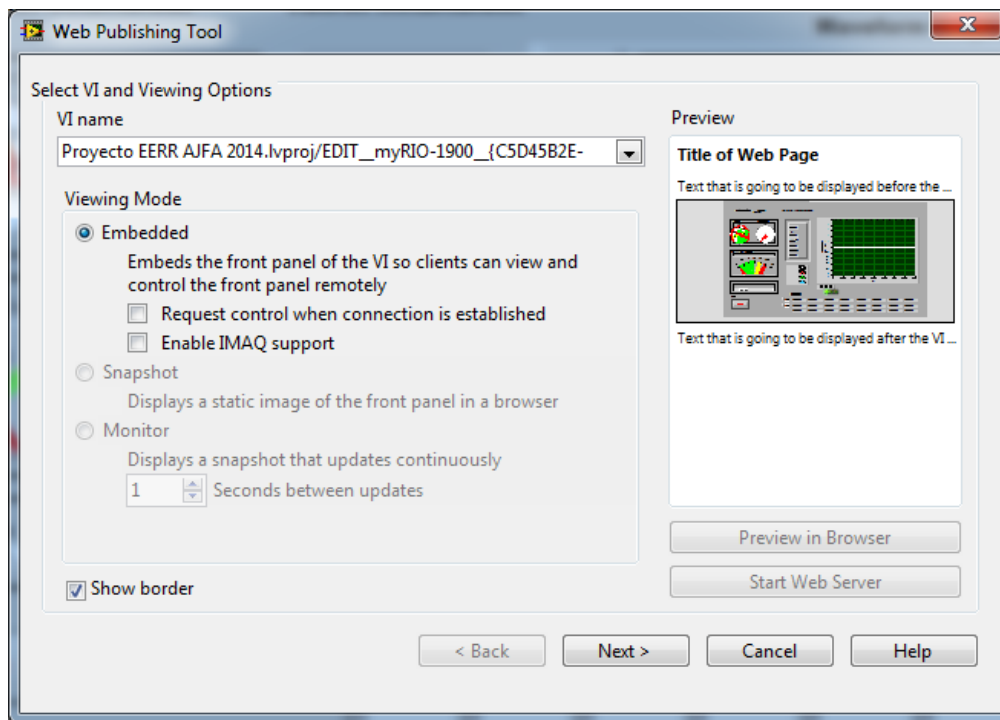


Figura 42: Opciones Control Web Publishing

- El sistema embebido que te permite actuar sobre el panel frontal, siendo este el más común.
- Snapshot, solo te permite visualizar el panel frontal y se actualiza este cada vez que actualices tu explorador.
- La tercera opción también es para visualizar solo, y le puedes programar un tiempo para que se actualice.

Le damos a next, le ponemos el nombre que queramos, y donde se va a guardar y la dirección URL, y se guarda en disco. La dirección de las publicaciones suele ser en la carpeta www, C:\Program Files (x86)\National Instruments\LabVIEW 2013\www

Importante deben estar en el mismo servidor.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

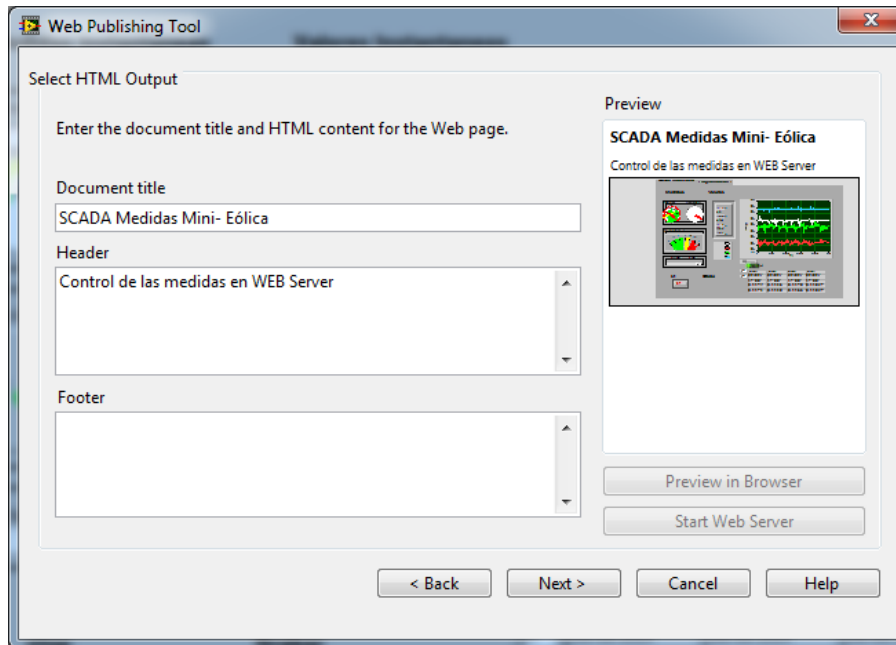


Figura 43: Nombre del Control Web Publishing

Por último guardamos la publicación con el nombre y la URL pertinente.

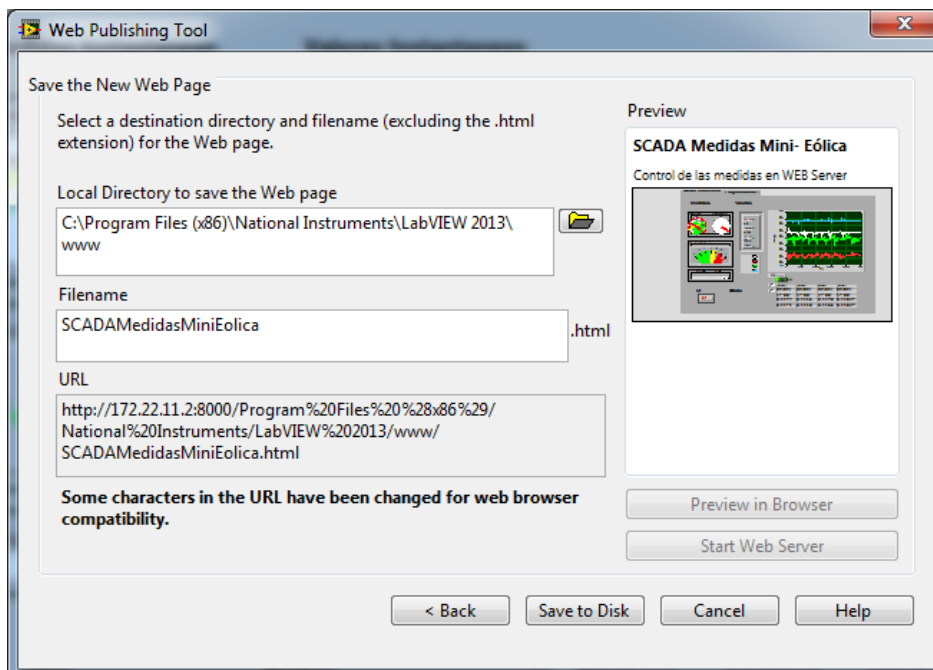


Figura 44: Configuración de ubicación Web



Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

### 6.6.2 Comunicación mediante un correo electrónico.

Voy a mandar un correo para indicar que el SCADA está parado o está averiado. Para ello Labview tiene una función dentro del grupo Data Communication > Protocols > SMTP

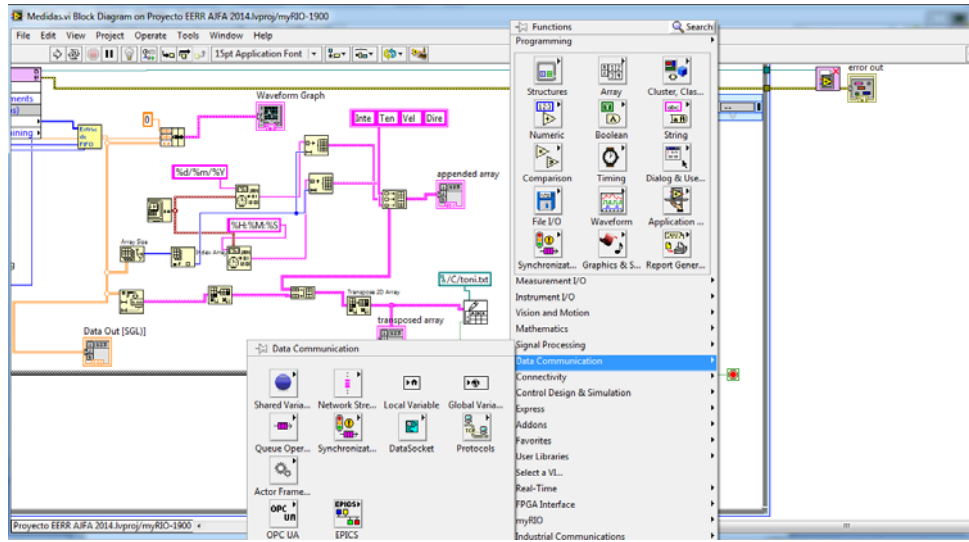


Figura 45: Esquema Data comunicacion

Inserto la función enviar email, y además lo uno al error para que lo enví.

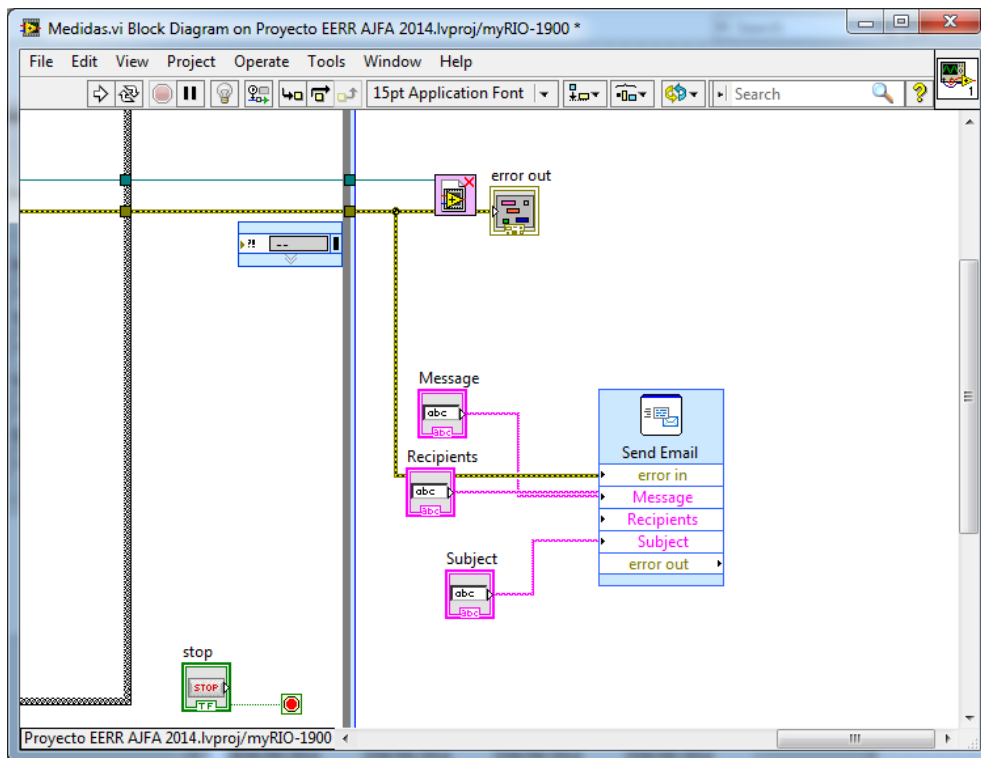


Figura 46: Insertar función enviar email

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

Por último indico los parámetros para enviar el correo, como indica en la figura siguiente

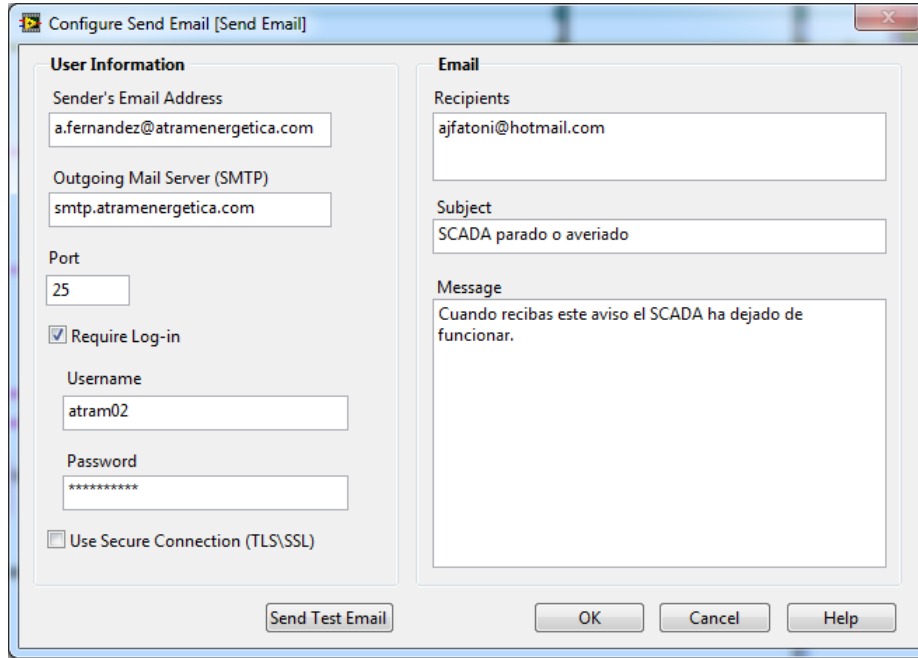


Figura 46: Parámetros de enviar email

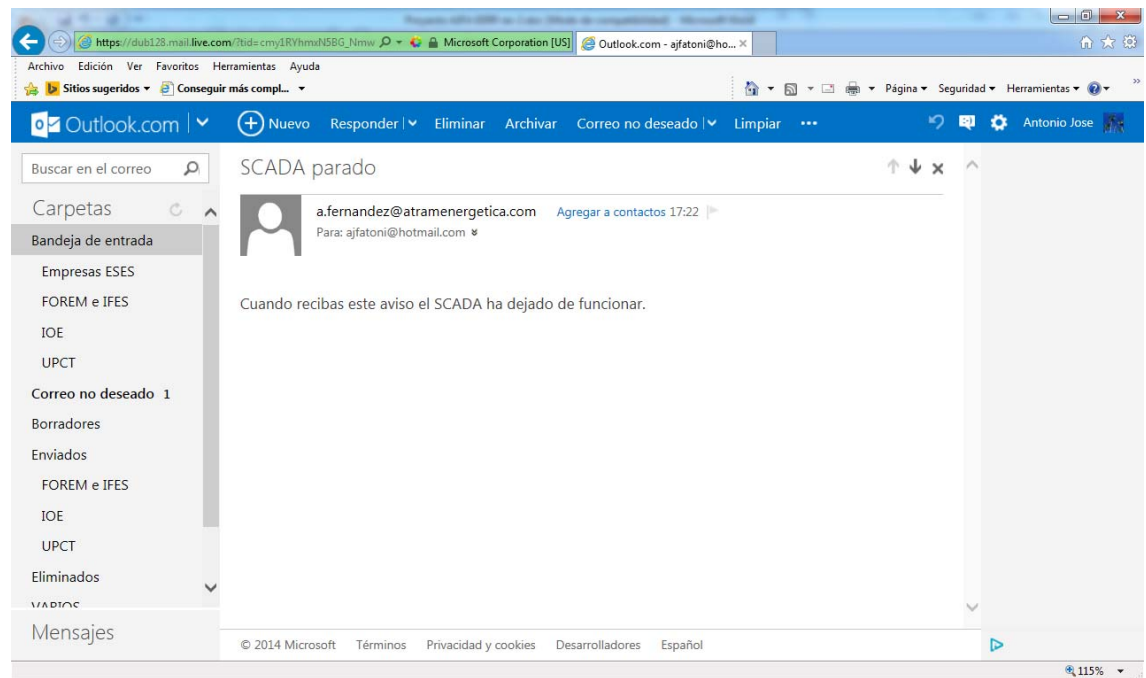


Figura 47: Comprobación de correo.

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.

## 7. Resultados.

### 7.1 Preparación ensayo .

Para simular las 4 señales a medir vamos a emplear el propio dispositivo, según las indicaciones del punto 6.3 Simulaciones de variables a medir.

Y tomaremos como parámetros de medida los que vemos en la siguiente figura:

- Numero de variables: 4 ud
- Muestras por variable: 100 ud
- Frecuencia de muestreo 100000 ms.
- Tiempo de descarga 5 sg:
- Time loop: 1000 ms.
- Time out de la FIFO: 5000 ms

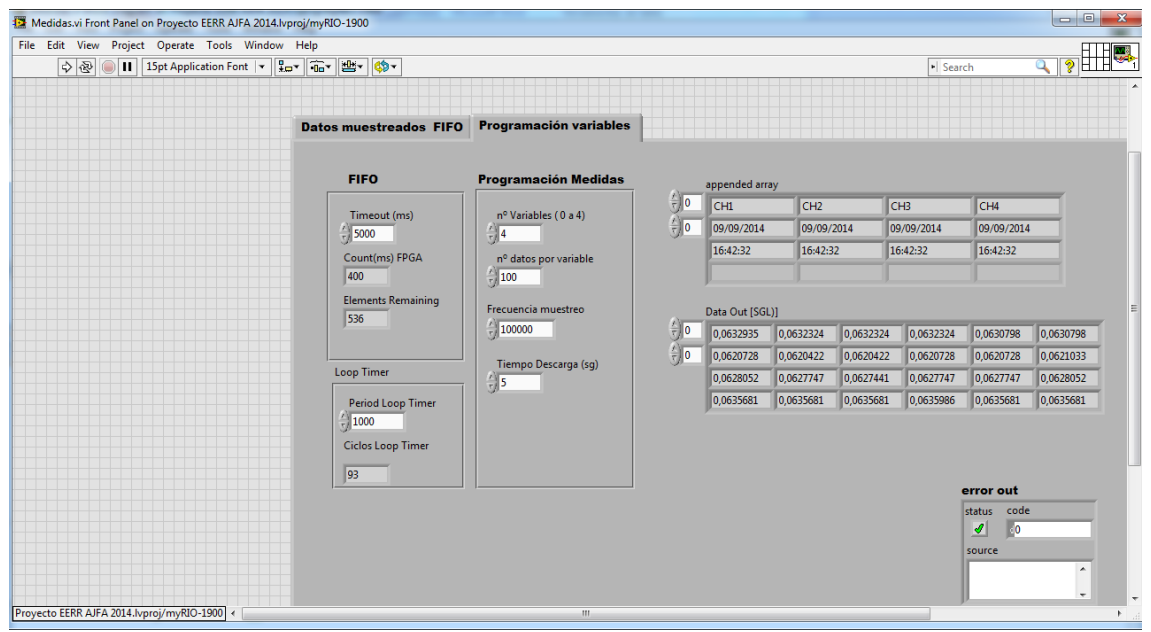
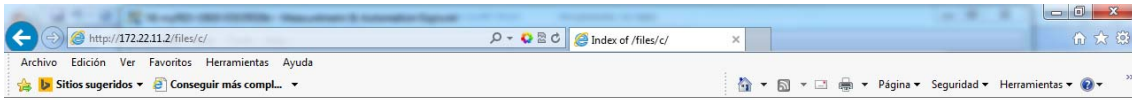


Figura 48: Vista parámetros de medida

### 7.2 Resultados.

Para recuperar los datos del ensayo debemos entrar en MAX y dentro del MyRIO, pulsar botón derecho y File Transfer, y nos aparece el disco. Te metes en c:

Aplicación de controladores comerciales para almacenamiento y monitorización remota de variables eléctricas.



## Index of /files/c/

- [Parent Directory](#)
- [Datos con FIFO.txt](#)
- [EERRAJFA.txt](#)
- [README File Paths.txt](#)
- [README File Transfer.txt](#)
- [ni-rt/](#)
- [toni.txt](#)
- [toni33.txt](#)

Embedthis Appweb 4.1.0 at Port 80

Figura 49: Ubicación de los datos de medida.

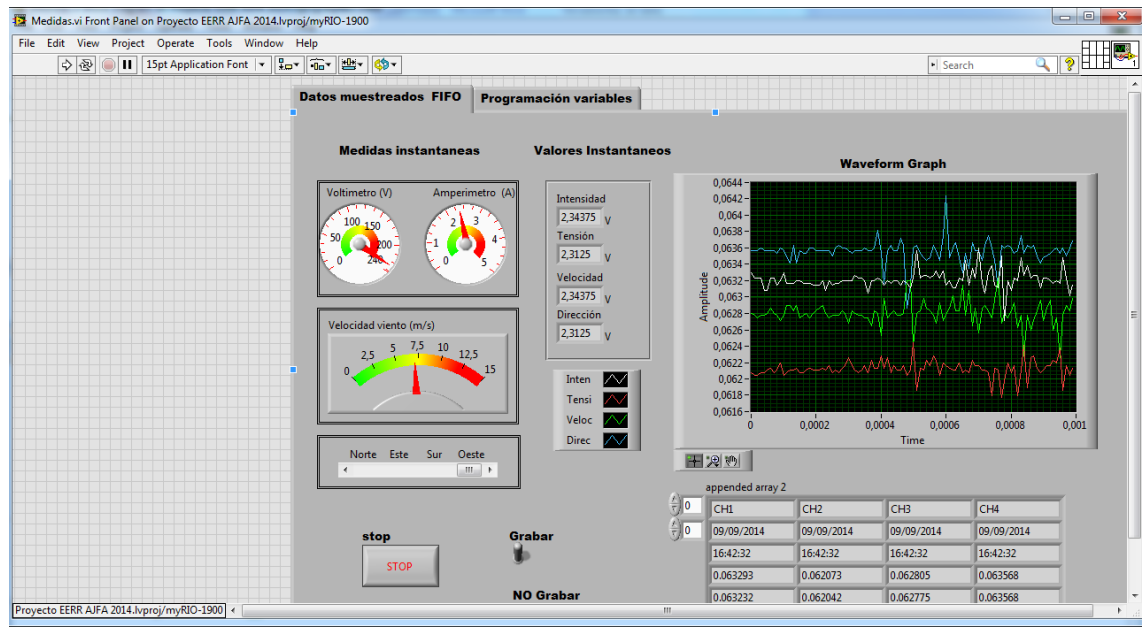


Figura 50: Vista parámetros de medida en panel frontal principal

Aplicación de controladores comerciales para almacenamiento y monitorización remota de avariables eléctricas.

CH1	CH2	CH3	CH4
09/09/2014		09/09/2014	09/09/2014
17:00:52		17:00:52	17:00:52
0.117432		0.115631	0.117004
0.117523		0.115875	0.116669
0.117493		0.115692	0.116913
0.117767		0.115356	0.116882
0.117523		0.115845	0.116913
0.117523		0.115540	0.117004
0.117523		0.115753	0.116882
0.117310		0.115601	0.117004
0.117493		0.115723	0.117188
0.117462		0.115692	0.116974
0.117462		0.115936	0.116455
0.117493		0.115692	0.117188
0.117493		0.115753	0.116882
0.117462		0.115723	0.117035
0.117523		0.115479	0.116974
0.117493		0.115723	0.117004
0.117249		0.115601	0.116974
0.117401		0.115692	0.117310
0.117432		0.115692	0.117035
0.117401		0.115692	0.117004
0.117493		0.115692	0.117096
0.117401		0.115784	0.116791
0.117493	0.115692	0.117065	0.118347

---

Figura 51: Vista parámetros de medida en panel frontal principal

## **8. Conclusiones y trabajos futuros.**

Como se ha observado se ha cumplido el objetivo del proyecto, medida de variable a alta frecuencia, con control remoto y comunicación de datos a través de correo.

Observamos que con el software Labview y el Hardware serie RIO de National Instrument disponemos de una configuración sencilla, flexible, económica y reconfigurable de nuestros parámetros de medida que nos permiten integrar y controlar las variables eléctricas y mecánicas de nuestros generadores mini eólicos.

Todo este desarrollo nos permitirá integrar con más facilidad la energía mini eólica en el mercado, ya que se desarrollara sistemas de medida y control económicos.

Respecto al coste de inversión material de la implantación de este sistema de medida rondaría aproximadamente unos 400 €/ud para la parte del Hardware, sin incluir el software, que estaría aproximadamente por unos 4.000 Euros.

Gracias a la gran flexibilidad que ofrece la serie Rio, 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.

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.

## **9. Bibliografía.**

- Molina Martínez J.M. y Jiménez Buendía M. Programación grafica para Ingenieros. Marcombo, 2010.
- Labview. Entorno Gráfico de Programación. Jose Rafael Lajara. Jose Pelegrí. Marcombo 2011.
- Real-Time Module User Manual. 2009.
- FPGA Module User Manual. Labview.2004.
- NI LabVIEW High- Performance FPGA Developer's Guide. Recommended Practices for Optimizing LabVIEW RIO Applications.
- Guia de desarrollo de NI para CompactRIO de National Instrument 2012.