

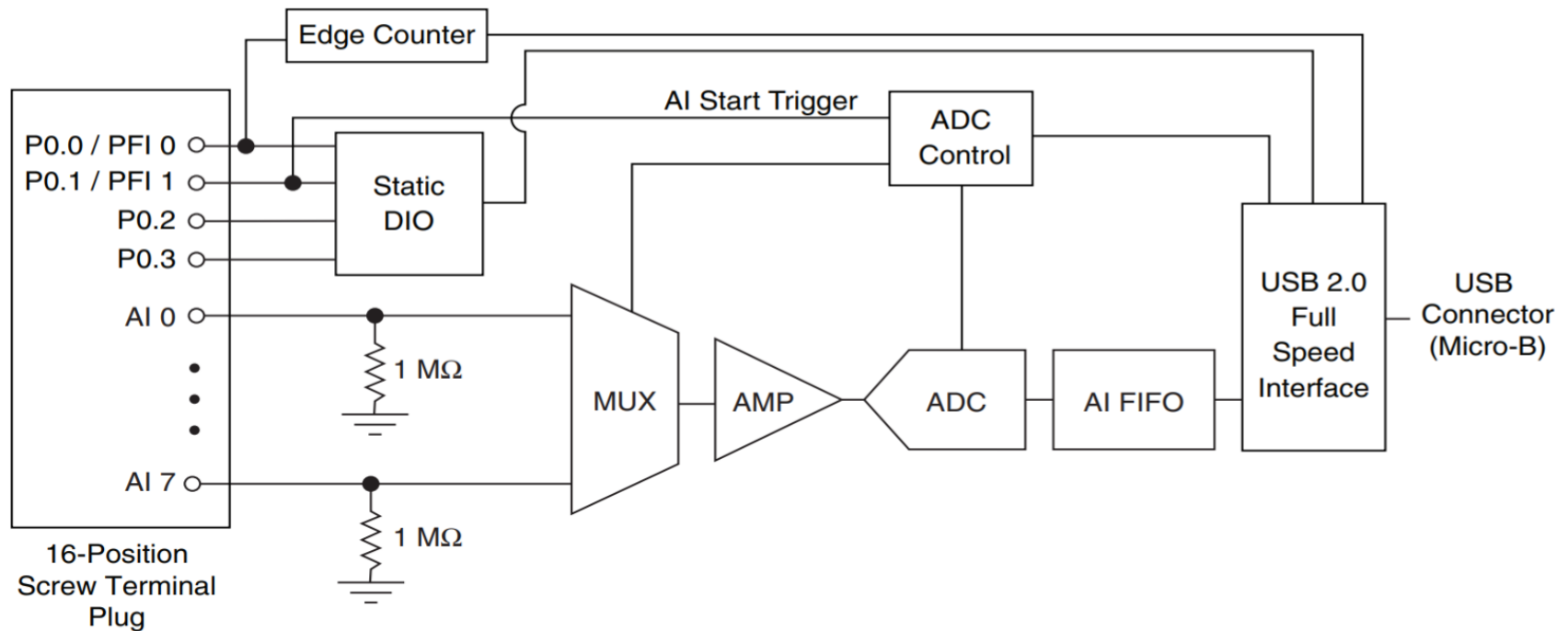
# Sistemas de Adquisición de Datos (DAQ) y acondicionamiento de la señal

Almudena Díaz Zayas

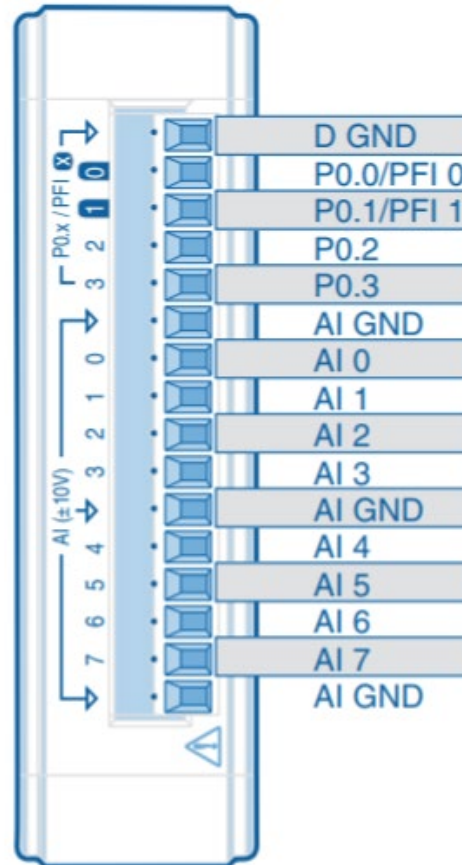
# NI USB-6000

# NI USB-6000 Diagrama de bloques

## Programmable Function Interface (PFI)



# NI USB-6000 Pinout



# USB – 6000 Specifications: Analog Input

Number of analog inputs	8, single-ended
Input resolution	12 bits
Maximum sample rate (aggregate), system-dependent	10 kS/s
Converter type	Successive approximation
AI FIFO	2,047 samples
Timing resolution	125 ns (8 MHz timebase)
Timing accuracy	100 ppm of actual sample rate
Input range	$\pm 10$ V
Working voltage	$\pm 10$ V
Input impedance	$>1$ M $\Omega$

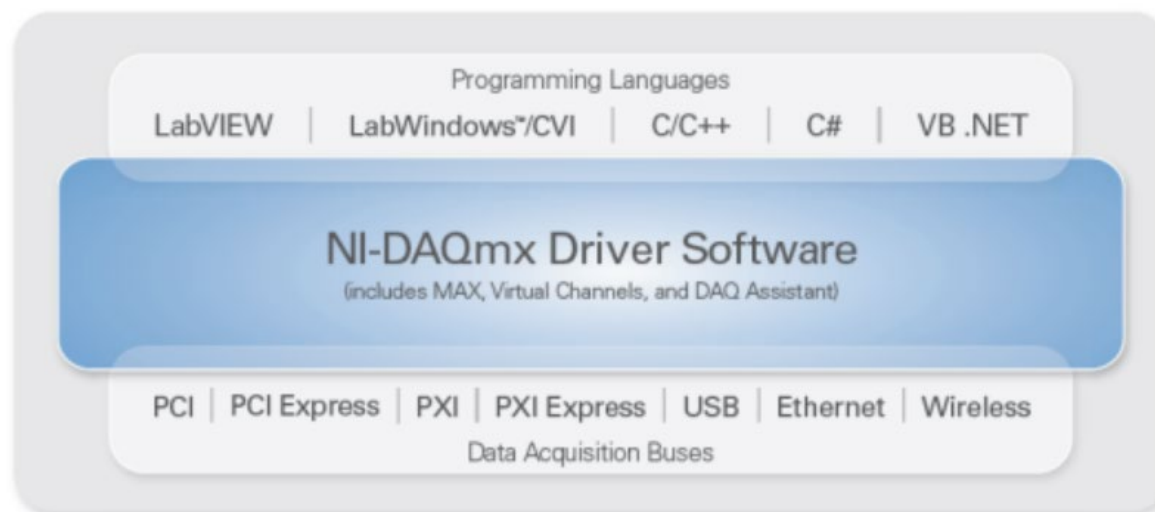
# USB – 6000 Specifications: Analog Input

Overvoltage protection	$\pm 30$ V
Trigger sources	Software, PFI 1
System noise <sup>1</sup>	10 mVrms
Absolute accuracy at full scale, single-ended	
Typical at 25 °C	26 mV
Maximum over temperature	135 mV

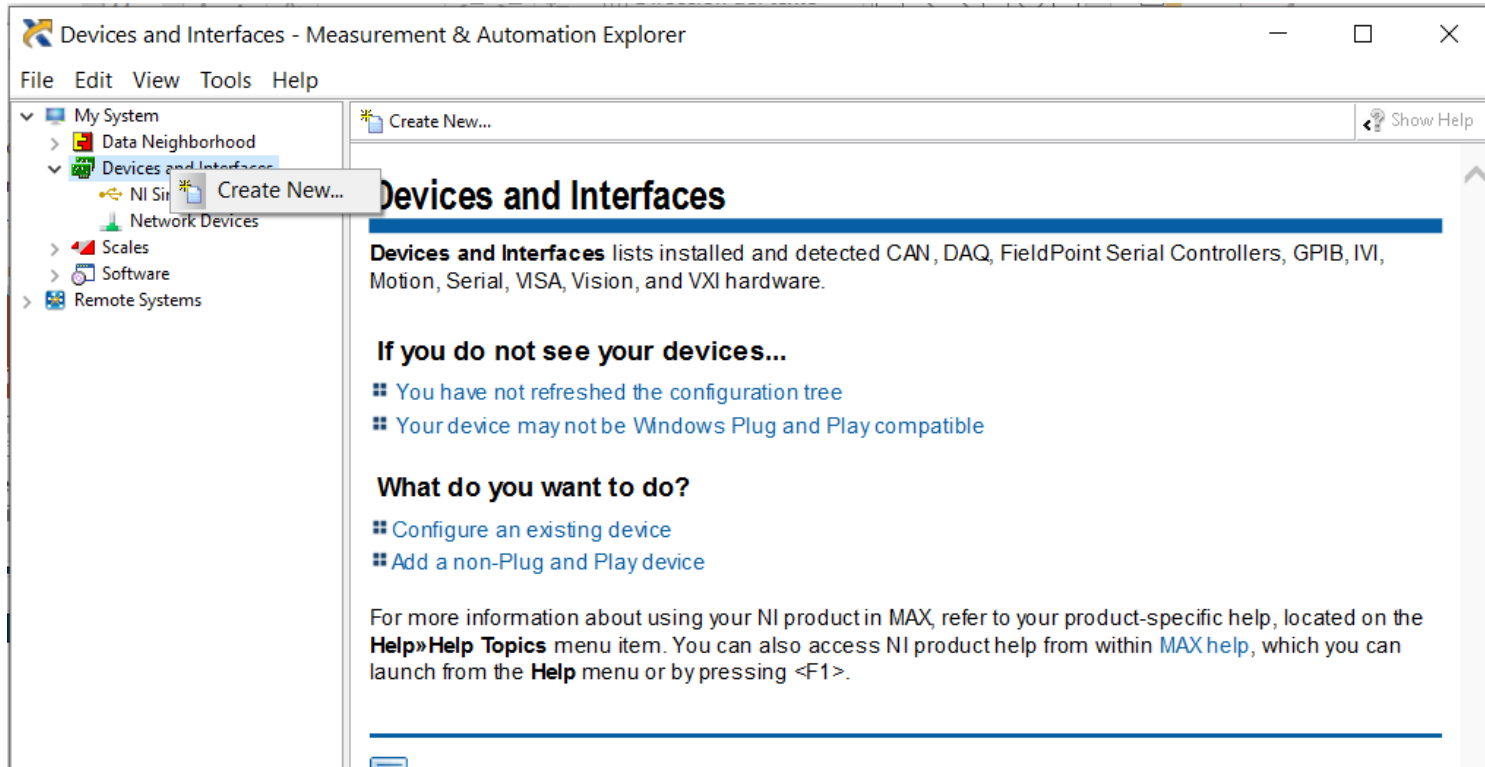
# USB – 6000 Specifications: Digital Input

Number of digital I/O	4
<b>Function</b>	
P0.0/PFI 0	Static digital I/O or counter source
P0.1/PFI 1	Static digital I/O or AI Start Trigger
P0.2	Static digital I/O
P0.3	Static digital I/O
Direction control	Each channel individually programmable as input or output
Output driver type	Each channel individually programmable as open collector or active drive
Absolute maximum voltage range	0 V to 5 V with respect to D GND
Pull-down resistor	47.5 k $\Omega$ to D GND
Power-on state	Input
<b>Digital Input</b>	
<b>Input voltage range</b>	
Powered on	0 V to 5 V
Powered off	0 V to 3.3 V
Input voltage protection	$\pm$ 20 V, for up to 24 hours

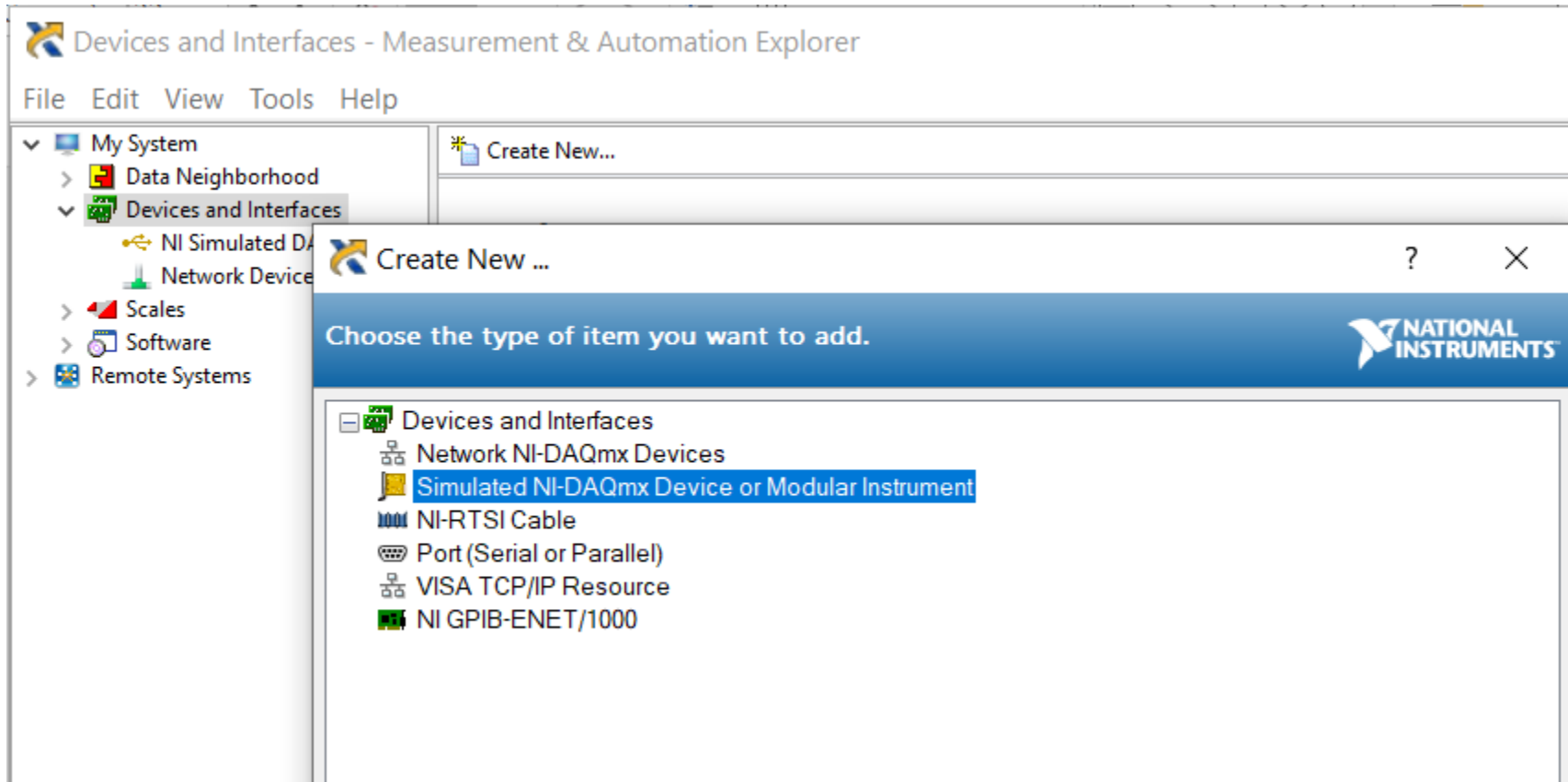
# NI-DAQmx



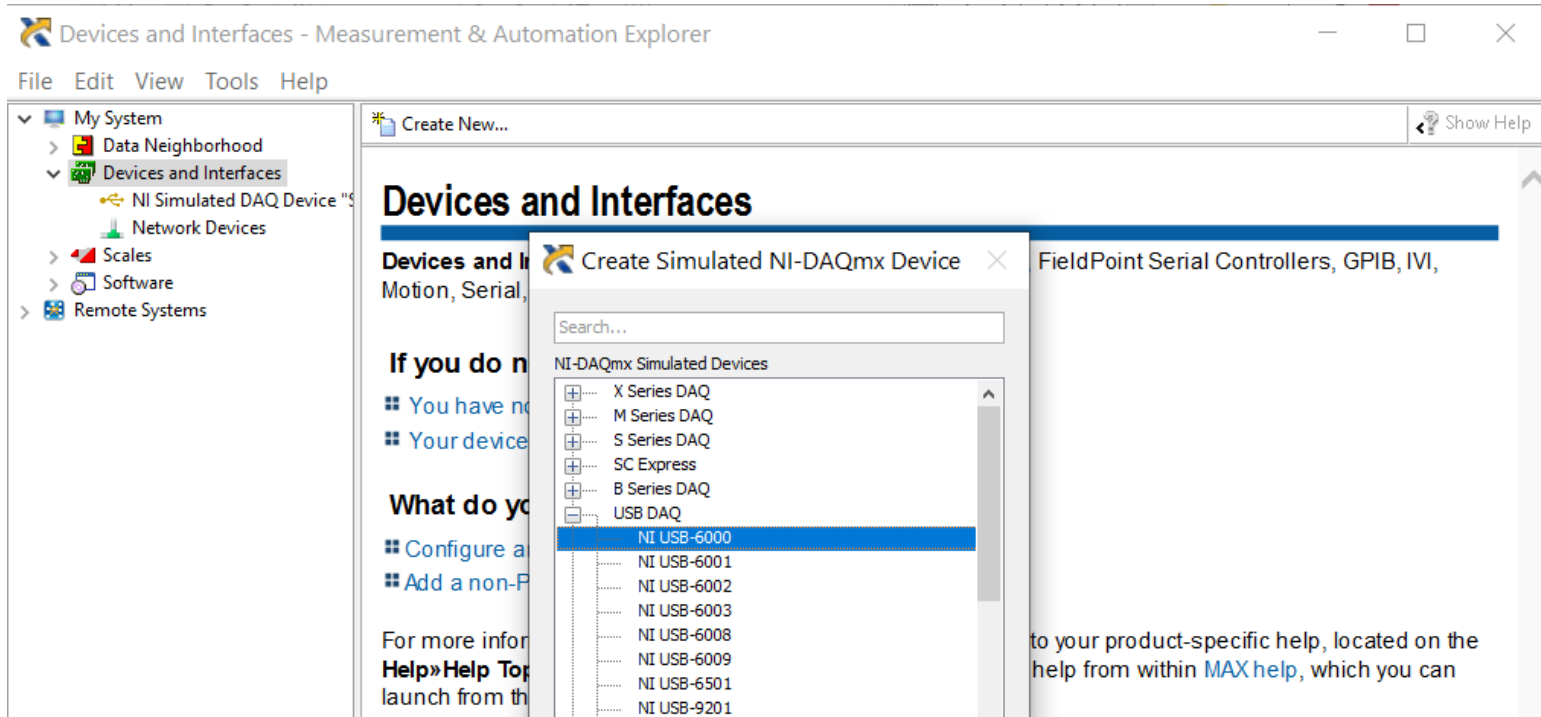
# Crear dispositivos simulados en NI-MAX

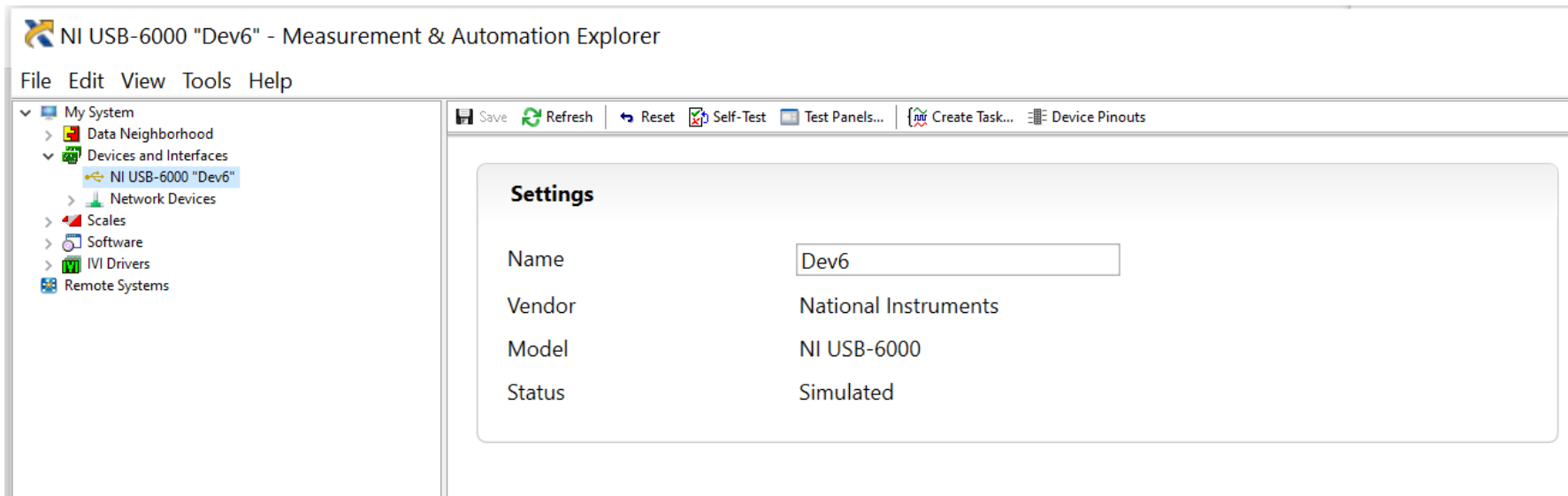


# Dispositivos simulados



# Dispositivos simulados





The screenshot shows the NI USB-6000 "Dev6" - Measurement & Automation Explorer interface. The window title is "NI USB-6000 'Dev6' - Measurement & Automation Explorer". The menu bar includes "File", "Edit", "View", "Tools", and "Help". The left sidebar shows a tree view with the following items: "My System", "Data Neighborhood", "Devices and Interfaces" (expanded), "NI USB-6000 'Dev6'" (selected), "Network Devices", "Scales", "Software", "IVI Drivers", and "Remote Systems". The main area displays the "Settings" for the selected device:

Settings	
Name	<input type="text" value="Dev6"/>
Vendor	National Instruments
Model	NI USB-6000
Status	Simulated

The toolbar at the top of the main area includes icons for "Save", "Refresh", "Reset", "Self-Test", "Test Panels...", "Create Task...", and "Device Pinouts".

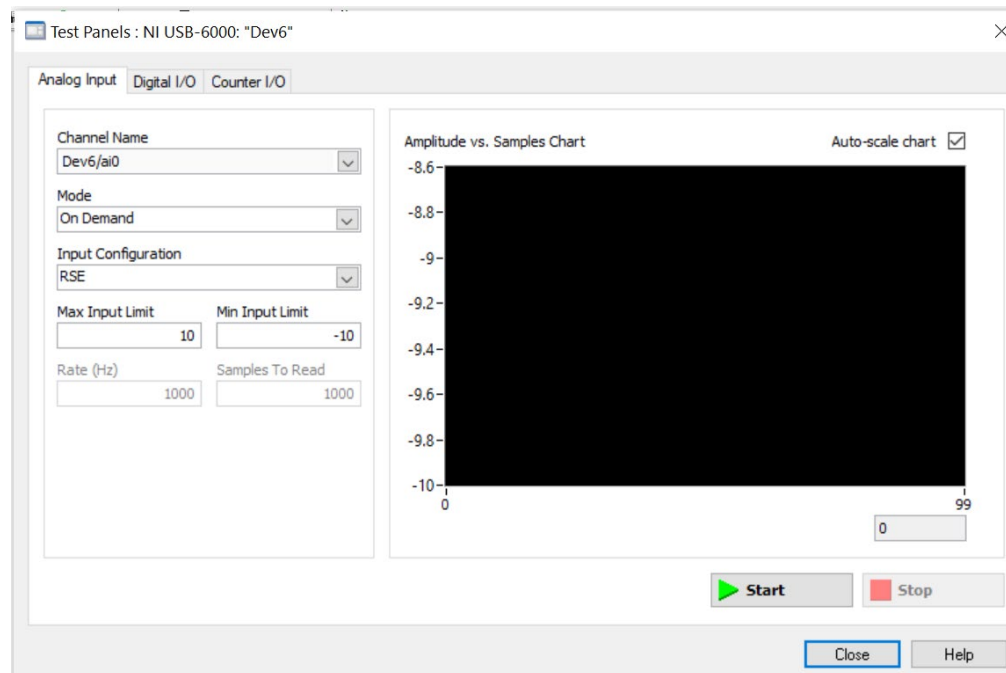
# MAX

The screenshot displays the NI USB-6000 "Dev6" - Measurement & Automation Explorer window. The interface includes a menu bar (File, Edit, View, Tools, Help) and a toolbar with icons for Save, Refresh, Reset, Self-Test, Test Panels..., Create Task..., and Device Pinouts. On the left, a tree view shows the system hierarchy: My System > Data Neighborhood > Devices and Interfaces > NI USB-6000 "Dev6". The main area shows the "Settings" for the selected device:

Settings	
Name	<input type="text" value="Dev6"/>
Vendor	National Instruments
Model	NI USB-6000
Status	Simulated

- NI-DAQmx es el driver que nos permite comunicarnos desde el PC con la tarjeta.
- NI-MAX software se usa para configurar y comprobar los dispositivos hardware.

# NI MAX: Test panels

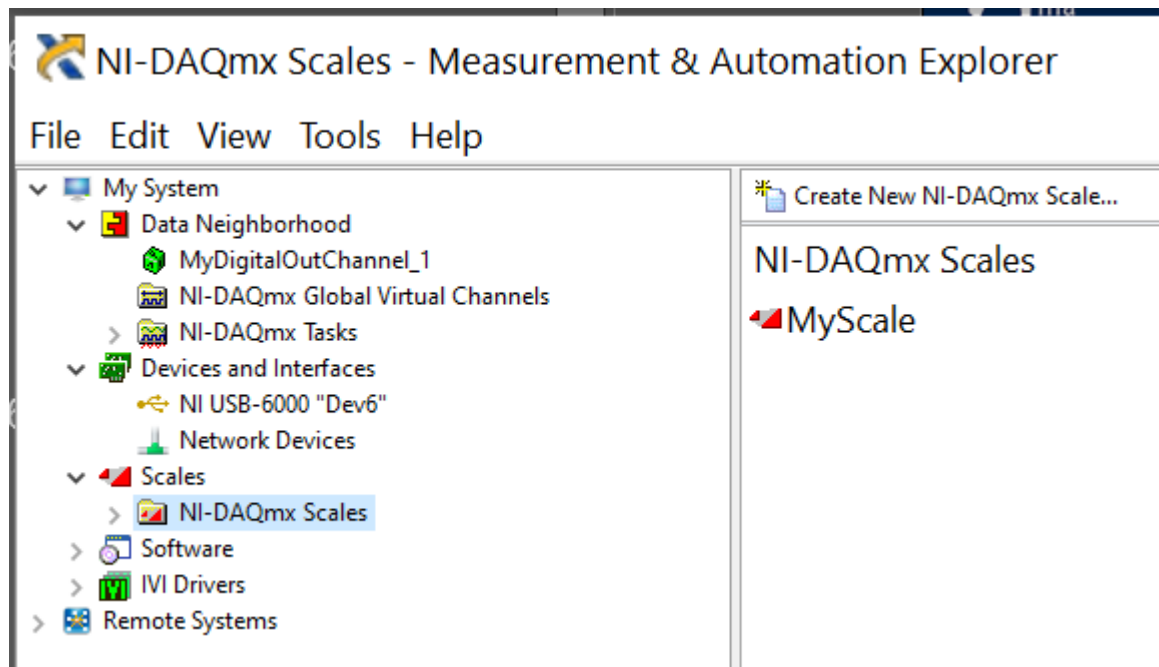


- **Channel name:** Se configura el canal en el cual se va a llevar a cabo la lectura
- **Input Configuration:**
  - **Differential:** modo diferencial para señales flotantes. El modo diferencial se usa para señales diferenciales, señales pequeñas o señales degradadas.
  - **RSE:** Medida referenciada a tierra.

- **Mode:**
  - **On demand:** Lectura de muestras de una en una usando una señal de reloj software
  - **Finite:** Se adquieren un número finito de muestras usando el reloj hardware interno
  - **Continuos:** adquiere un número ilimitado de muestras usando el reloj hardware interno

- **“Rate” and “Samples to read”**
  - Sólo se aplica en los modos “finite” y “continuous”.
  - **Rate:** Frecuencia de muestreo
  - **Samples to read:** Número de muestras totales durante una adquisición finita. En el caso de una adquisición continua se esperará a tener el número de muestras indicadas antes de realizar la representación.

# MAX: Scales



NI-DAQmx Scales - Measurement & Automation Explorer

File Edit View Tools Help

My System

- Data Neighborhood
  - MyDigitalOutChannel\_1
  - NI-DAQmx Global Virtual Channels
  - NI-DAQmx Tasks
- Devices and Interfaces
  - NI USB-6000 "Dev6"
  - Network Devices
- Scales
  - NI-DAQmx Scales
    - MyScale
      - Create New NI-DAQmx Scale...
- Software
- IVI Drivers
- Remote Systems

NI-DAQmx Scales

MyScale

Create New NI-DAQmx Scale...

Create New NI-DAQmx Scale...

NATIONAL INSTRUMENTS

Select the type of scale to use in the measurement.

Custom scales specify a conversion between a scaled value and a phenomenon a device measures or generates. For example, the pressure of an ideal gas in a closed container is related to its temperature. You can create a virtual channel to measure temperature and use a custom scale that converts that temperature to a pressure reading.

When using a custom scale in an application, specify the minimum and maximum value in terms of the scaled units. For input operations, the custom scale is used to convert the real world units into your scaled units. For output operations, the custom scale is used to convert your scaled units into real world units.

- Linear**—Scales values by using the equation  $y=mx+b$ , where  $x$  is a prescaled value and  $y$  is a scaled value.
- Map Ranges**—Scales values proportionally from a range of prescaled values to a range of scaled values.
- Polynomial**—Scales values using an  $n$ th order polynomial equation.
- Table**—Maps an array of prescaled values to an array of corresponding scaled values, with all other values scaled proportionally.

Linear

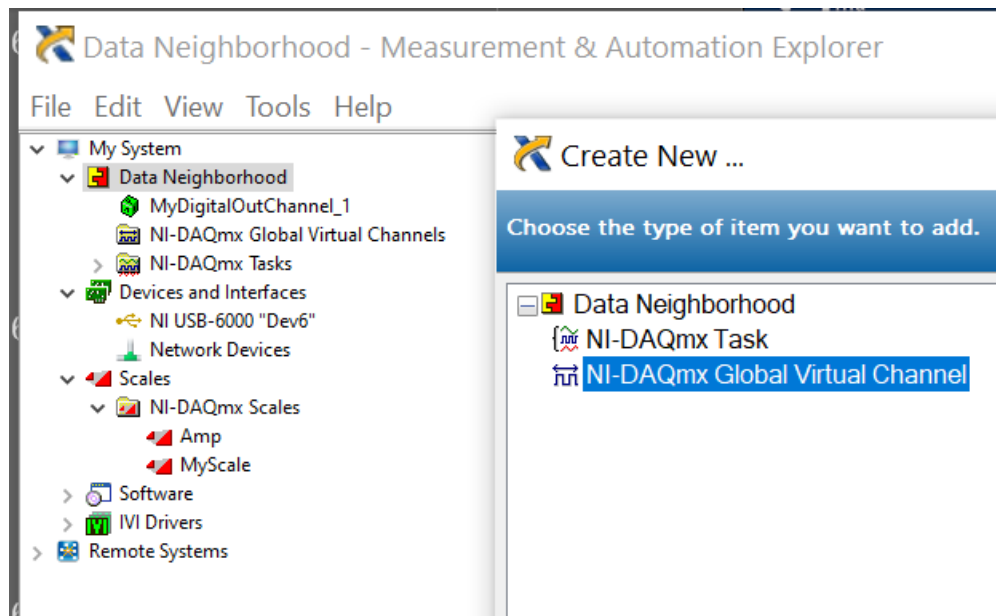
Map Ranges

Polynomial

Table

< Back Next > Finish Cancel

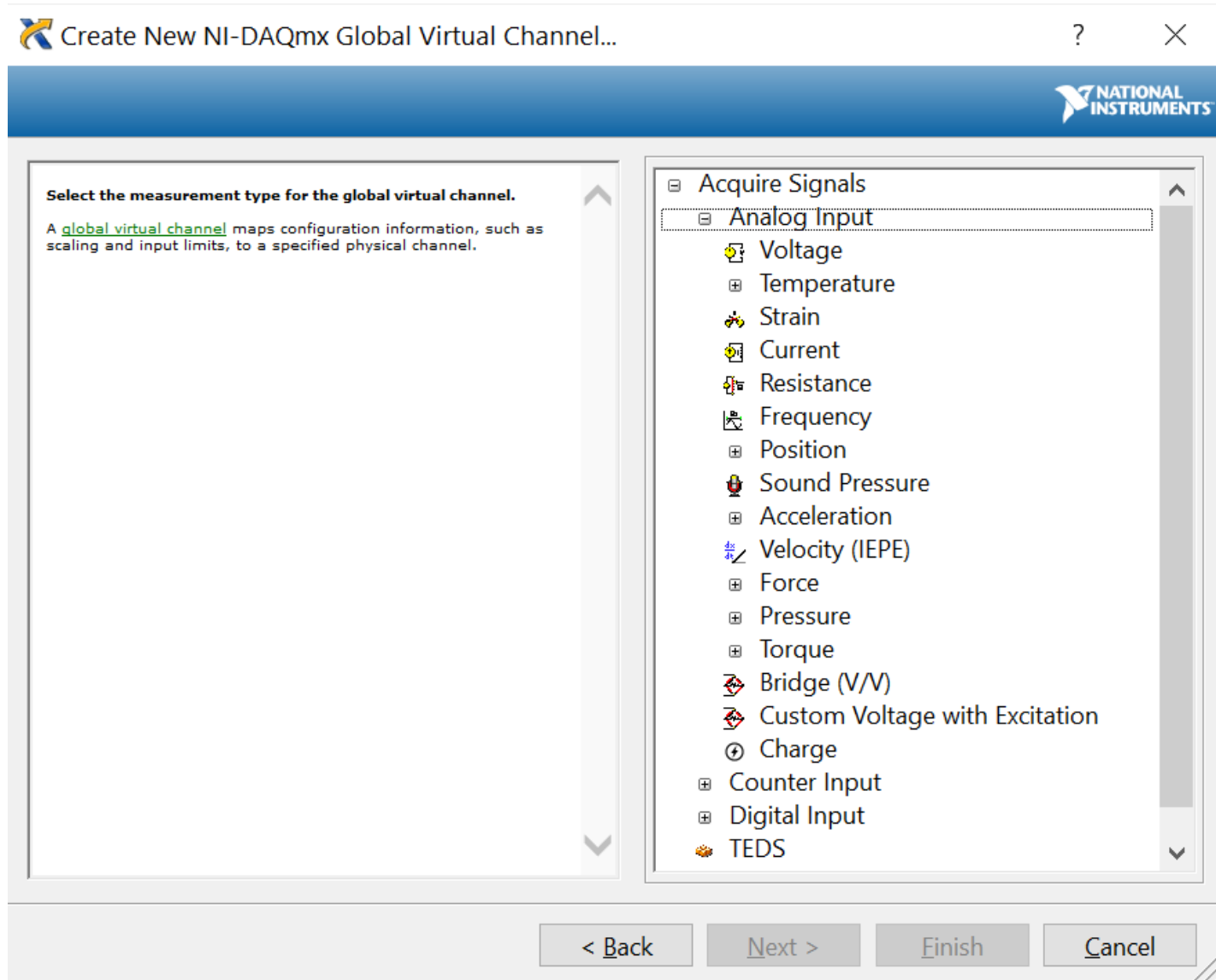
# MAX: Virtual channels



- **Virtual channels**

- Los canales virtuales globales son entidades de software que encapsulan el canal físico junto con otra información específica del canal, como rango, configuración de la terminal y escala personalizada.
  - Un canal virtual asigna información de configuración como la escala y los límites de entrada.

# Virtual channels



# Virtual channels

Create New NI-DAQmx Global Virtual Channel... ? X

**NATIONAL INSTRUMENTS**

**Select the physical channel to use for your new global virtual channel.**

If you have previously configured [global virtual channels](#) of the same measurement type, you can click the **Virtual** tab to add or copy the information from an existing global virtual channel.

If you have configured TEDS channels, you can click the **TEDS** tab to use a TEDS channel for your new global virtual channel.

**Physical**

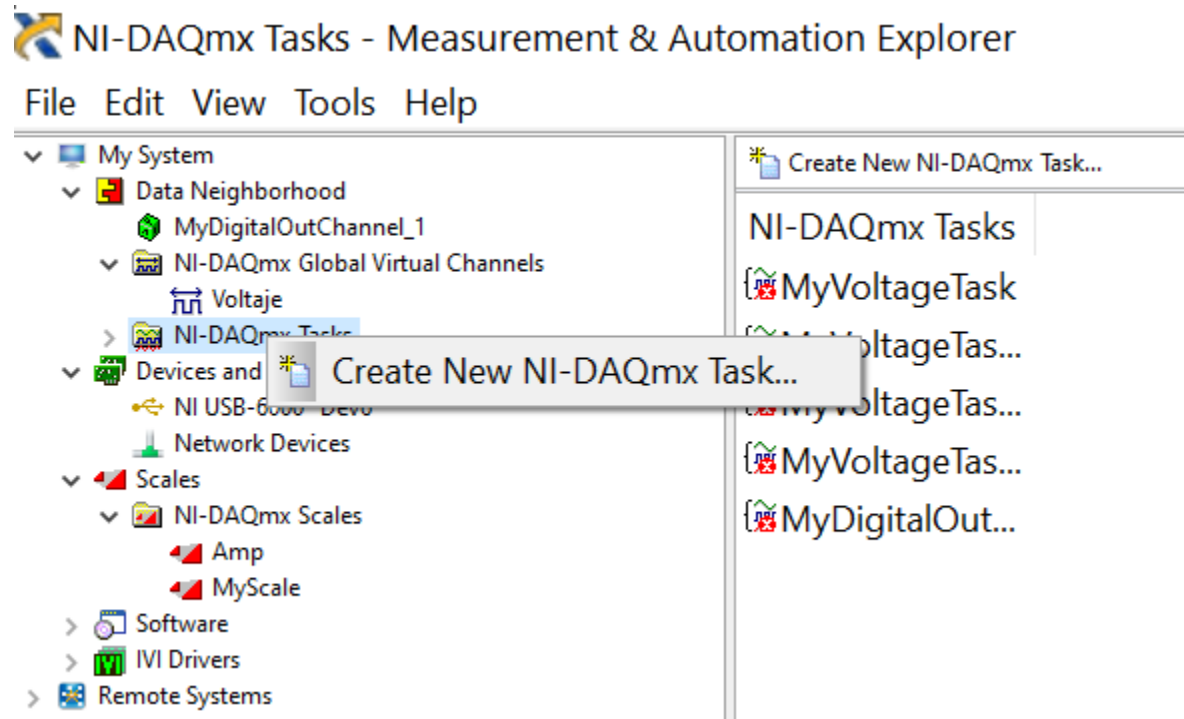
Supported Physical Channels

- [-] Dev6 (USB-6000)
  - ai0
  - ai1
  - ai2
  - ai3
  - ai4
  - ai5
  - ai6
  - ai7

< Back   Next >   Finish   Cancel




# MAX: Tasks



- **Tasks**

- Una tarea es una colección de uno o más canales, así como la temporización, el disparo y otras propiedades que aplican a la tarea.
- Una tarea es una colección de canales con propiedades similares.
- Los canales que componen la tarea pueden ser utilizados en múltiples tareas (canal global) o asignado a una tarea específica (canal local).

✚ Create New ...
?
✕



**Select the measurement type for the task.**


A **task** is a collection of one or more virtual channels with timing, triggering, and other properties.

To have **multiple measurement types** within a single task, you must first create the task with one measurement type. After you create the task, click the **Add Channels** button to add a new measurement type to the task.

- ☐ Acquire Signals
  - ☐ Analog Input
    - 🔌 Voltage
    - ⊕ Temperature
    - 🔧 Strain
    - 🔌 Current
    - 🔌 Resistance
    - 📊 Frequency
    - ⊕ Position
    - 🔊 Sound Pressure
    - ⊕ Acceleration
    - 📊 Velocity (IEPE)
    - ⊕ Force
    - ⊕ Pressure
    - ⊕ Torque
    - 🔌 Bridge (V/V)
    - 🔌 Custom Voltage with Excitation
    - ⊕ Charge
  - ⊕ Counter Input
  - ⊕ Digital Input
  - 🔥 TEDS

< Back
Next >
Finish
Cancel

✚ Create New ...
? ✕




**Select the global virtual channel(s) to add to the task.**


You can add existing [global virtual channels](#) or copy information from existing global virtual channels to virtual channels local to the task. Only global virtual channels that match the measurement type of the task appear.

Select **Add Global Channels** to add existing global virtual channels to the task.

Select **Copy Global Channels** to copy information from selected existing global virtual channels and create [local virtual channels](#) in the task.

For hardware that supports [multiple channels](#) in a task, you can select multiple channels to add to the task at the same time.





Physical
Virtual

Supported Global Channels

- [-] MAX Channels
  - [-] Voltaje

Add Global Channels

Copy Global Channels

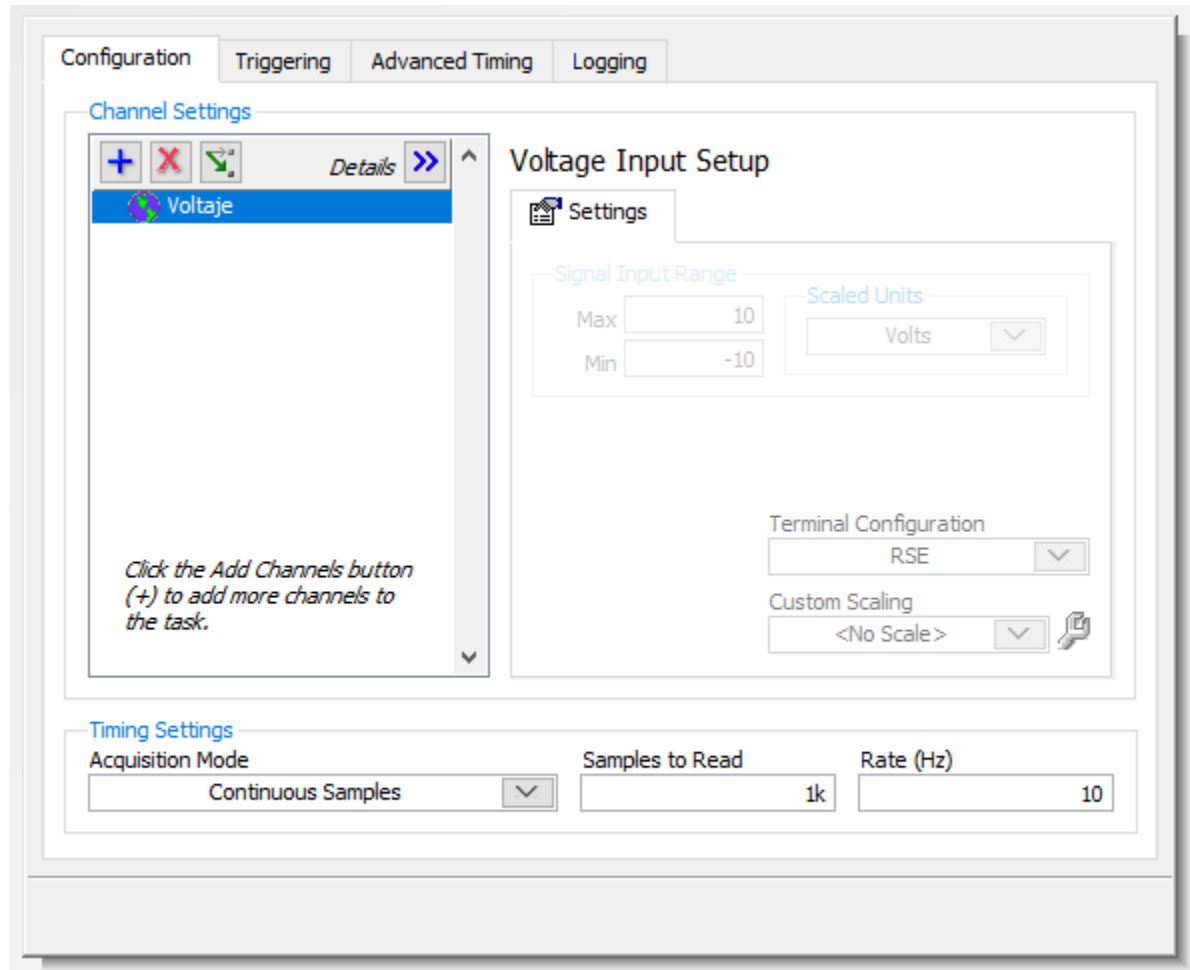
<Ctrl> or <Shift> click to select multiple channels.

< Back

Next >

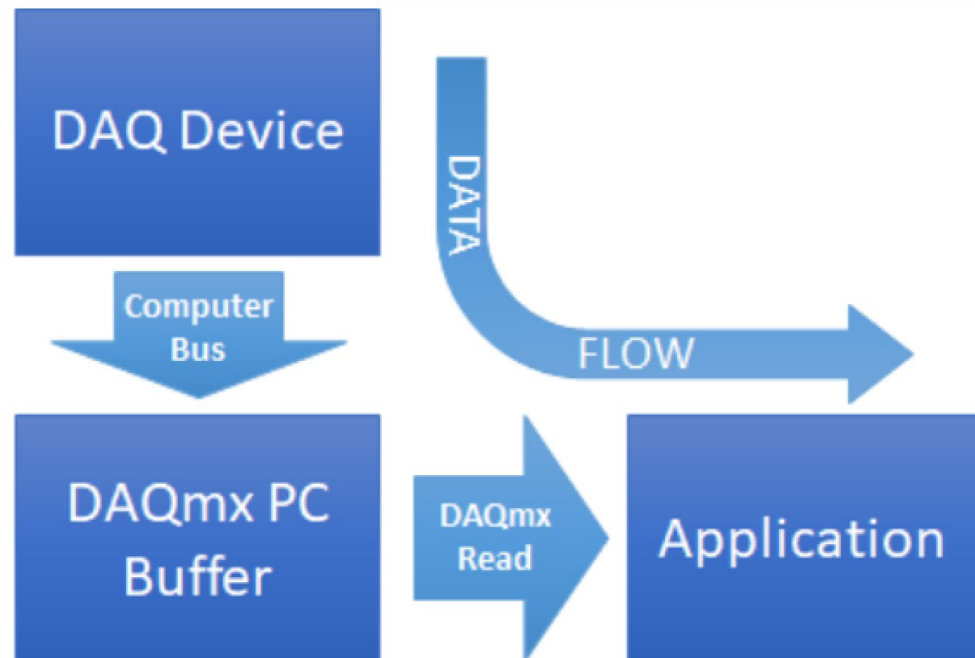
Finish

Cancel



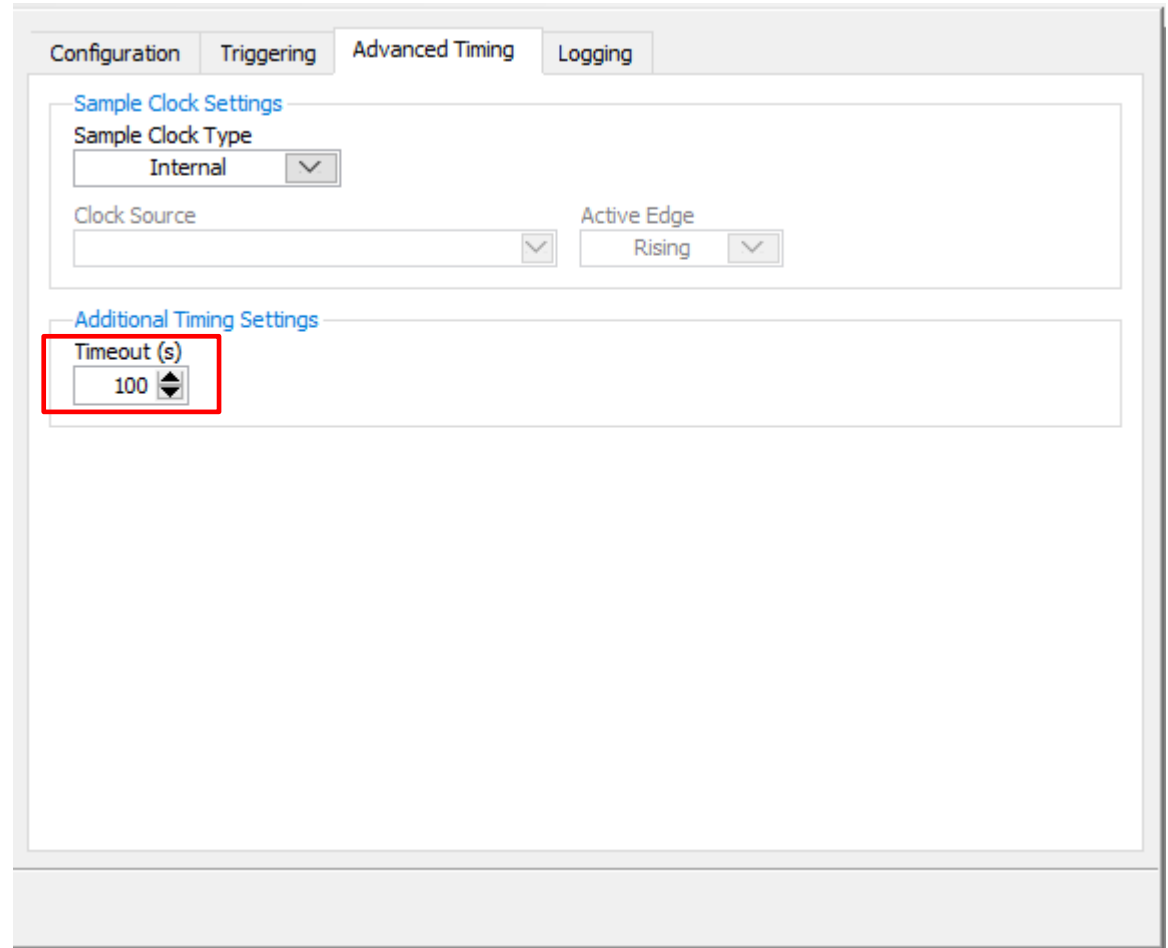
Al realizar una adquisición con DAQmx, los datos se transfieren primero desde el dispositivo DAQ a la memoria RAM del ordenador. En el ordenador hay un buffer preasignado en la memoria RAM llamado DAQmx PC Buffer. En este buffer se almacenan los datos no leídos desde el nivel de aplicación.

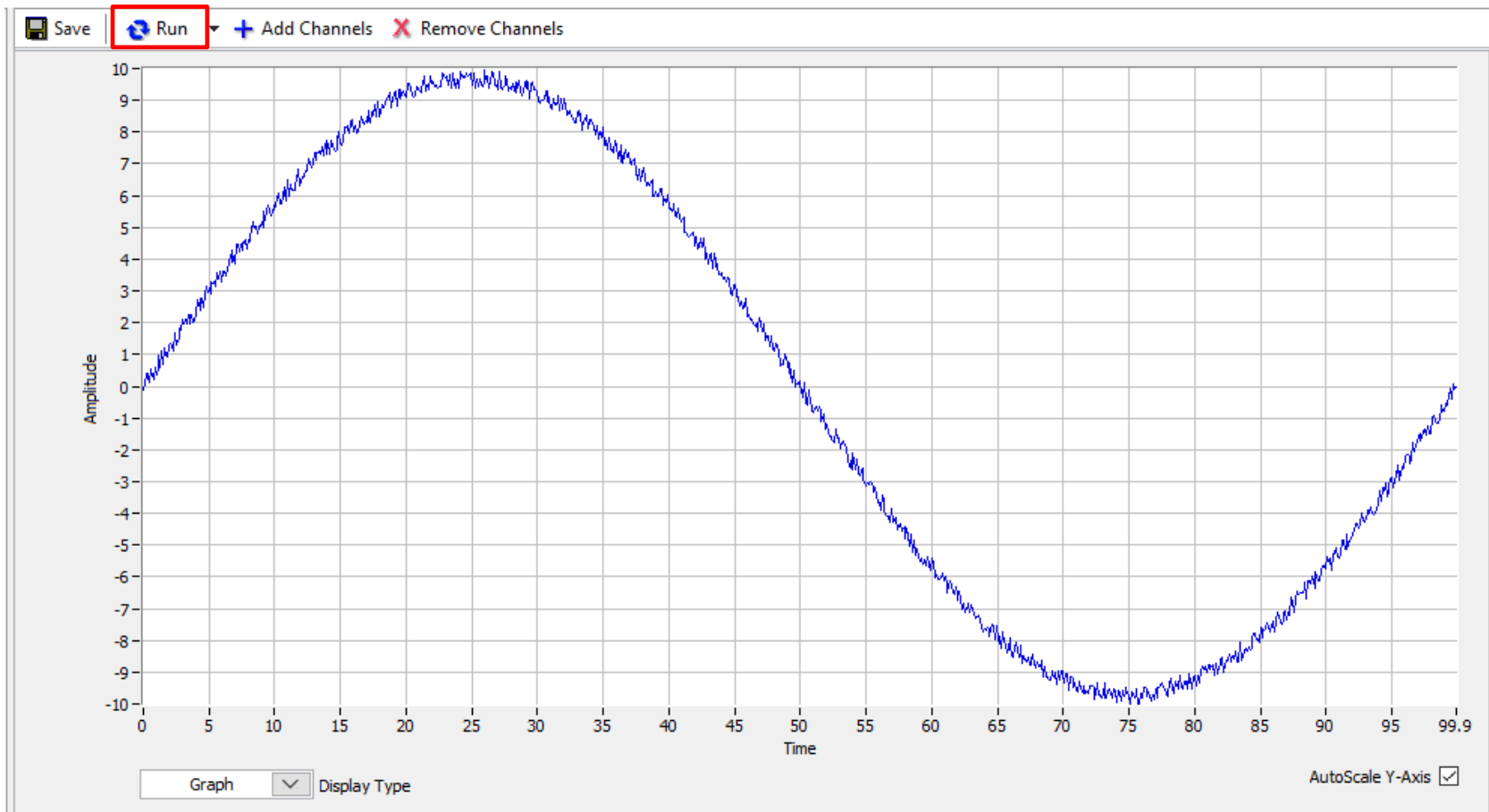
En **modo continuo** el valor de **Samples per Channel** se utiliza para determinar el tamaño del DAQmx PC Buffer si el tamaño del buffer predeterminado no es suficiente. Sin embargo, en la mayoría de los casos, se utiliza el tamaño de buffer predeterminado y se ignora este valor.



Especifica en segundos la cantidad de tiempo hasta que expira el temporizador.

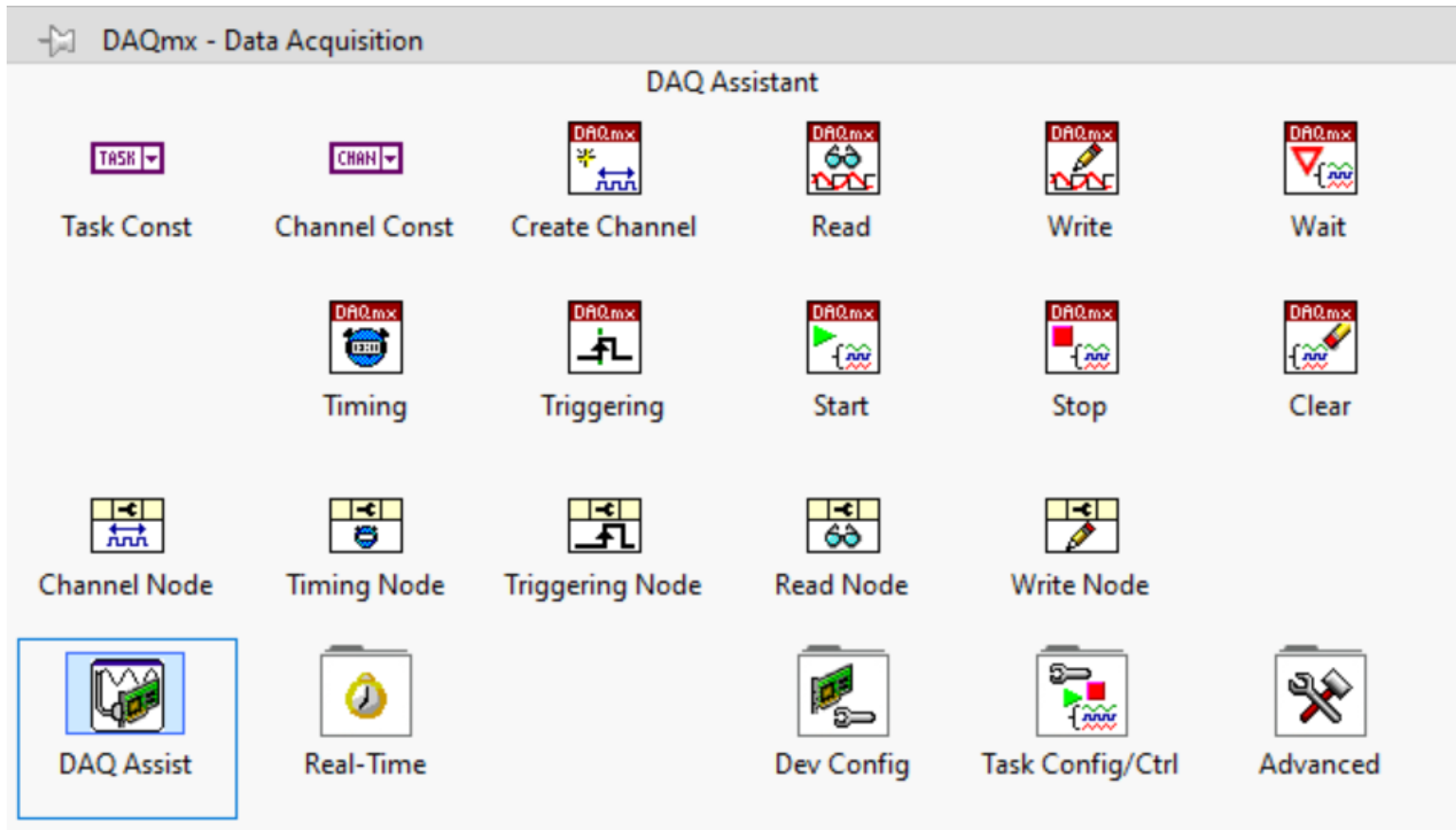
Un valor de -1 significa que el temporizador interno nunca expira. Se usa un -1 si se usa una señal de disparo y se quiere evitar que expire el temporizador.



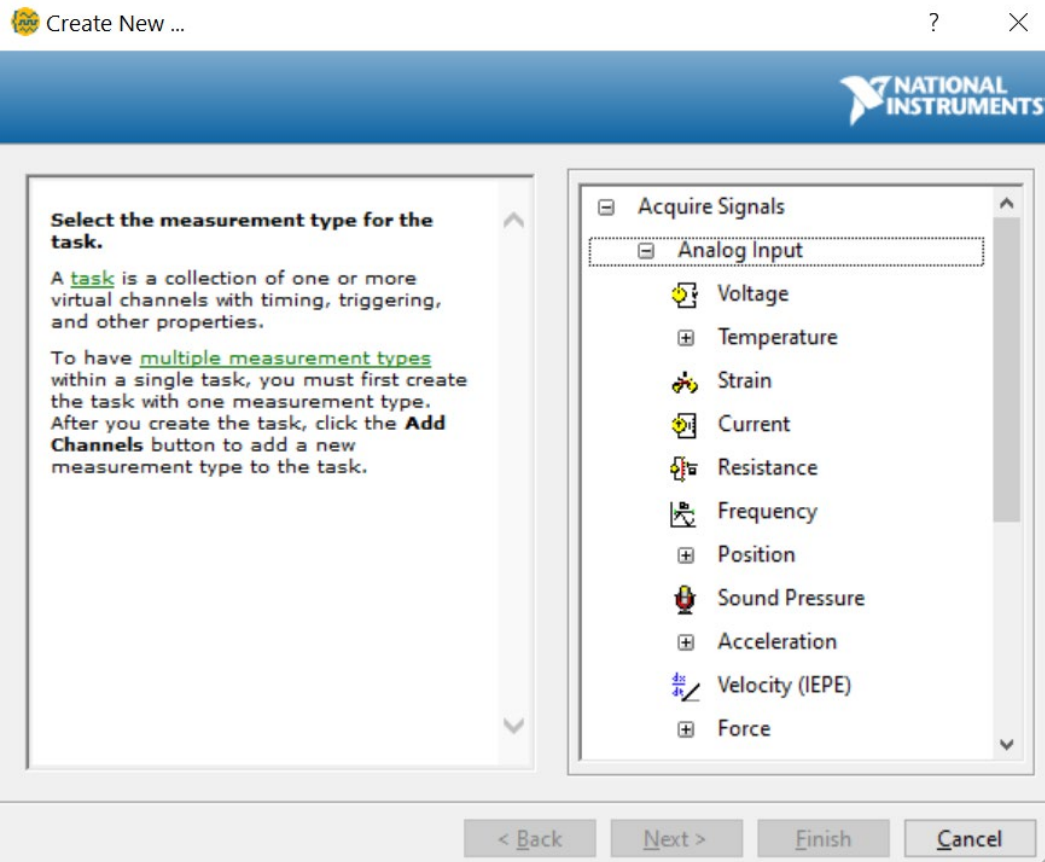


# Labview DAQ Assist

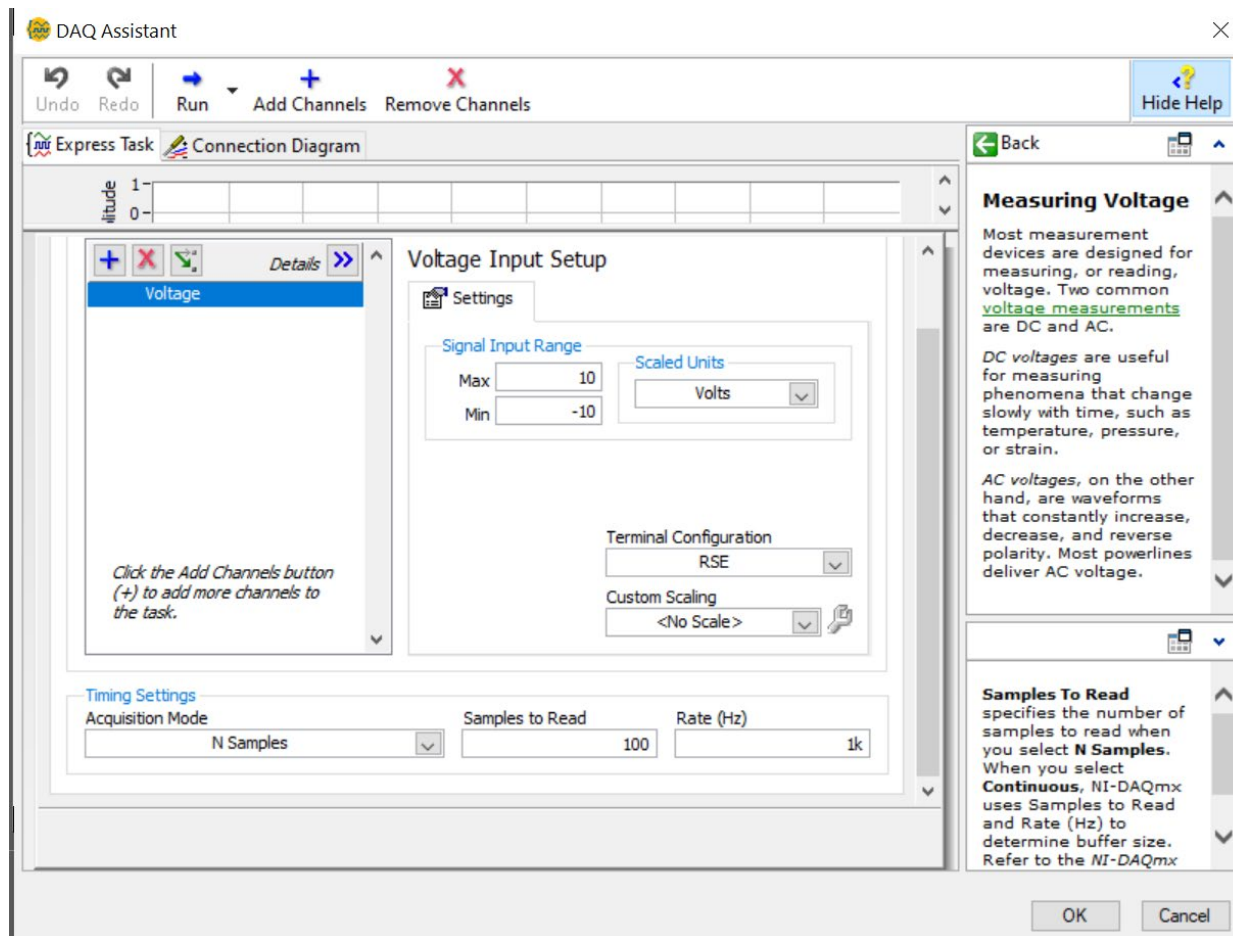
# Paleta Measurement I/O → DAQ Assist



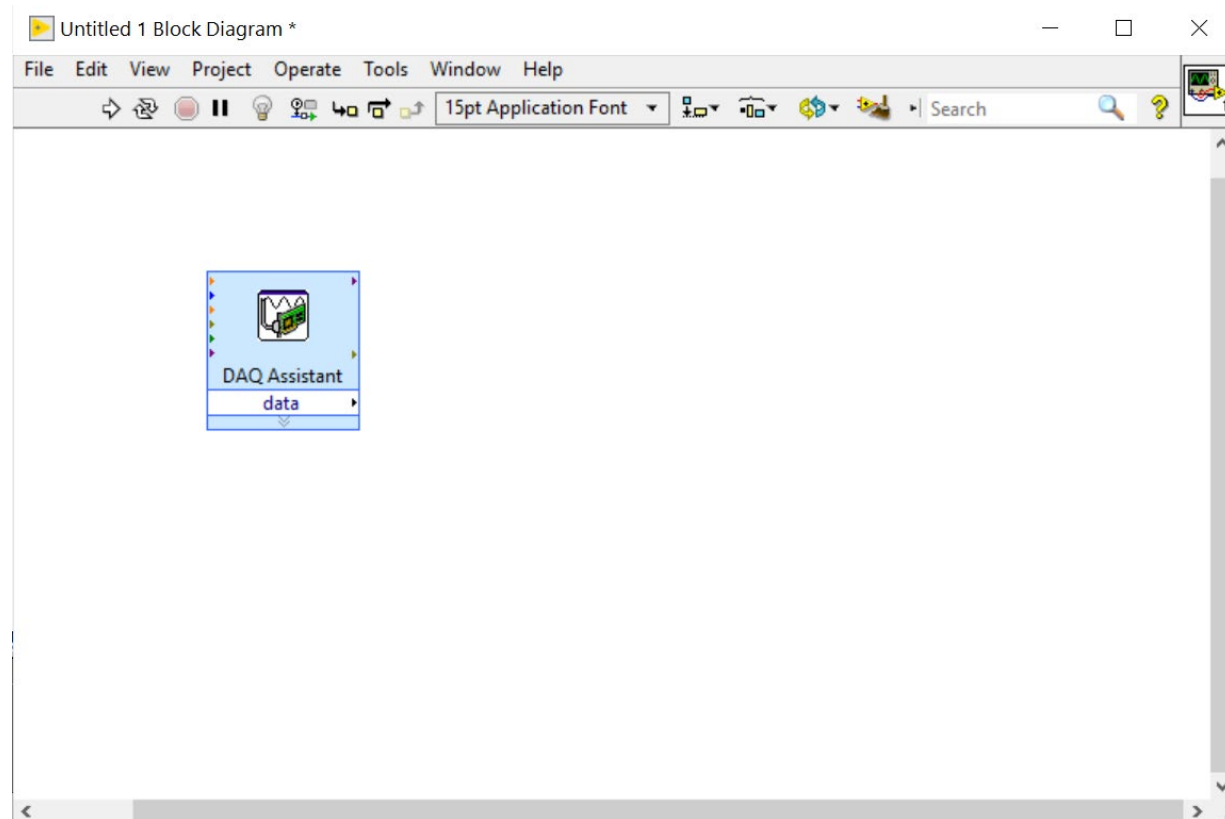
Este componente proporciona un cuadro de diálogo que permite seleccionar la adquisición o la configuración de una señal.



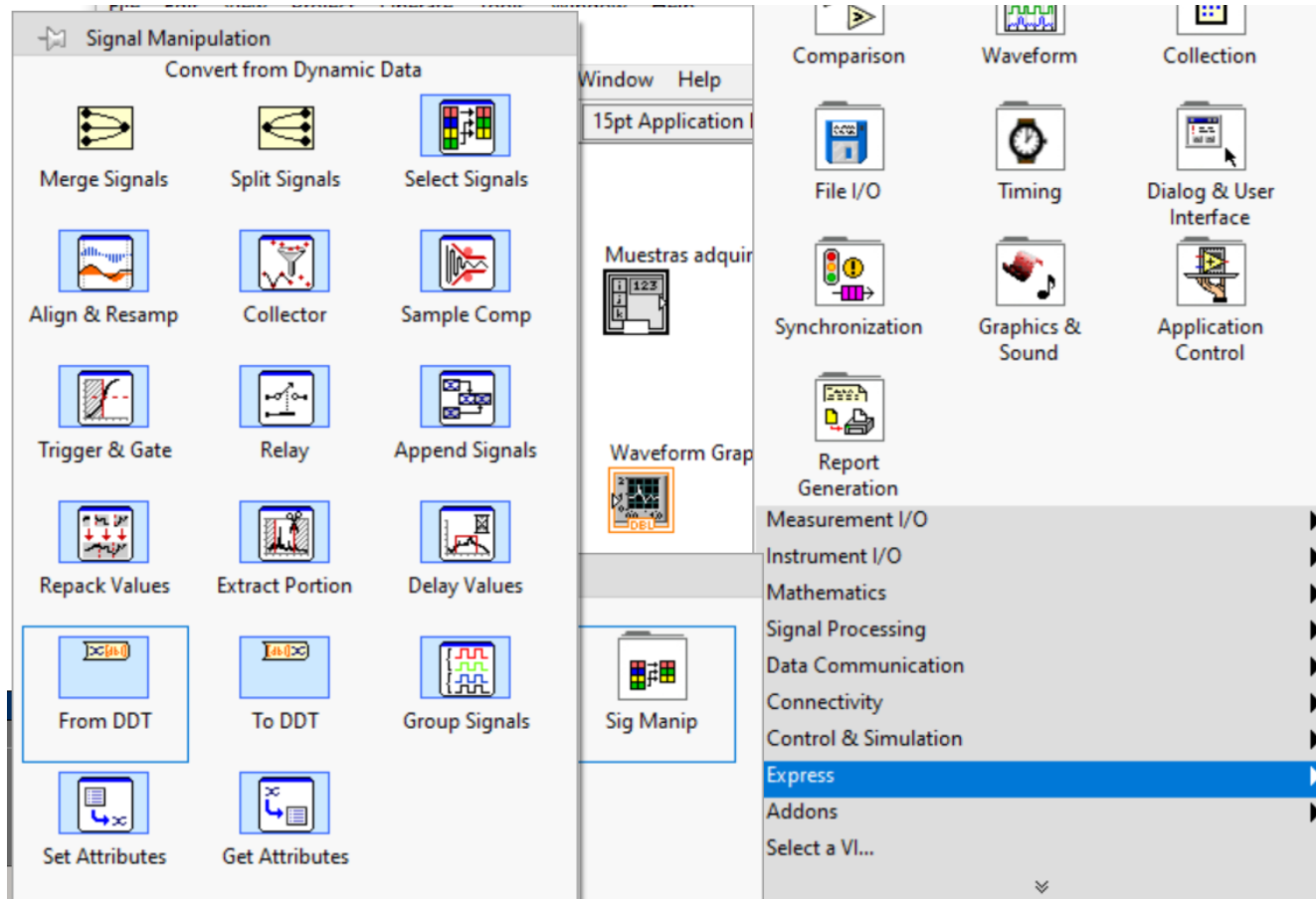
Una vez seleccionada la medida o generación de una señal, el tipo y el canal se muestra el cuadro de diálogo para configurar la tarea.



Una vez configurada la tarea se mostrará un mensaje de progreso mientras LabVIEW construye el DAQ Assistant VI. Una vez terminado el proceso el bloque presenta terminales de datos para la lectura o escritura de señales. El terminal de datos es de tipo dinámico.

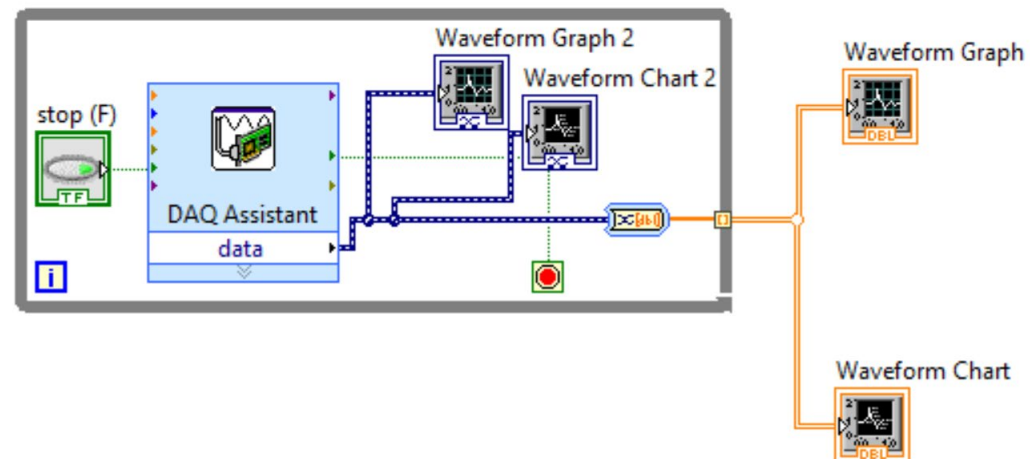
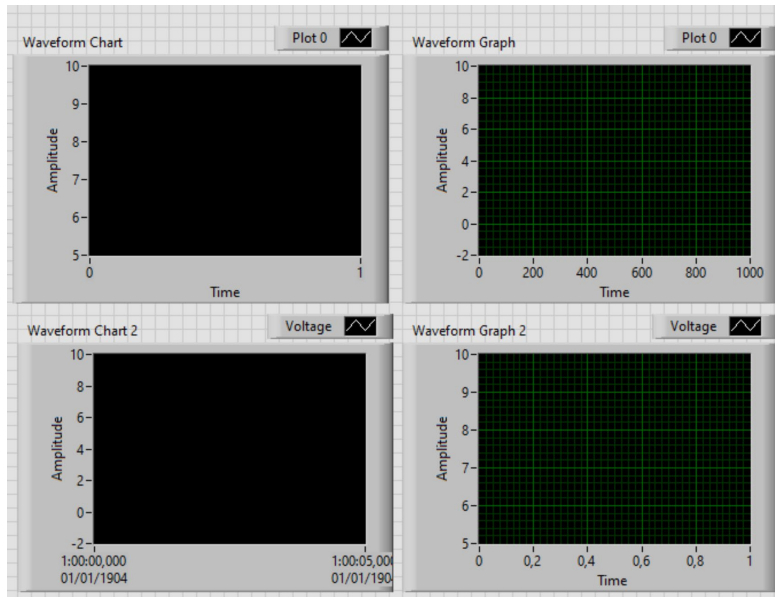


Usar el VI de la paleta Express “Convert from Dynamic Data” para convertir el tipo de dato dinámico en un escalar.



## Ejercicio DAQ Assist

- 1) Usar el asistente para configurar la adquisición de una señal de voltaje. Configurar el modo de adquisición "N Sample" en la tarea.
- 2) Construir el panel frontal mostrado en la figura.

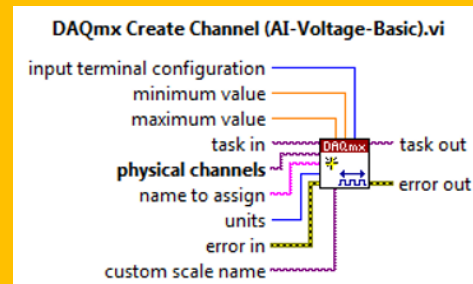


## Tareas

### Canales

Sensores,  
escalado,  
formato

Configuración  
del canal



Temporización, triggering

Encaminamiento de la señal

Configuración del búfer

Configuración hardware

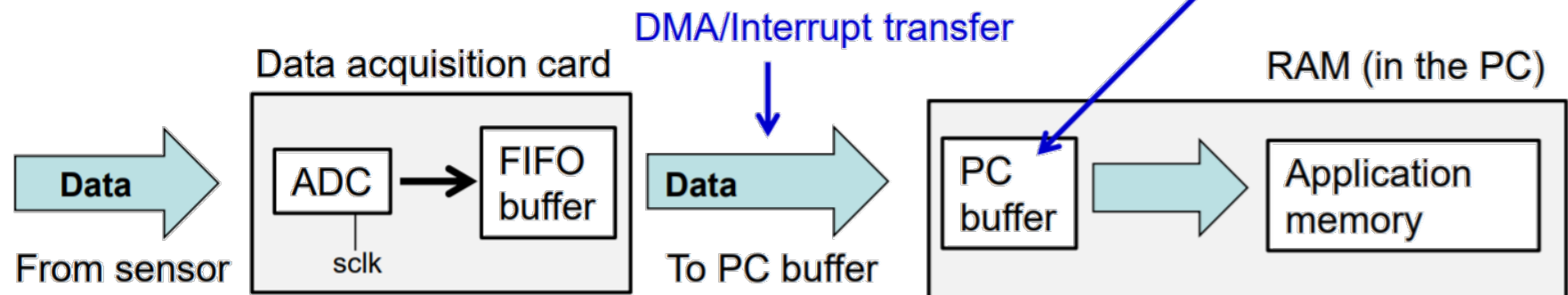
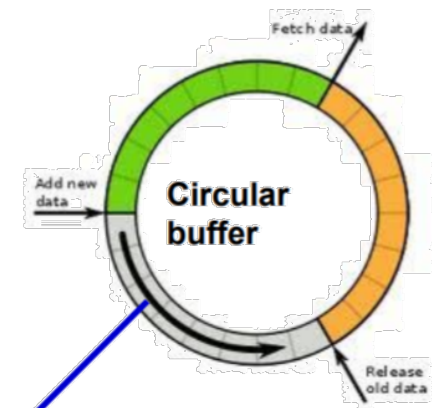
# Transferencia de datos desde la tarjeta DAQ

Los datos adquiridos se almacenan en el **búfer FIFO** hardware de la tarjeta DAQ.

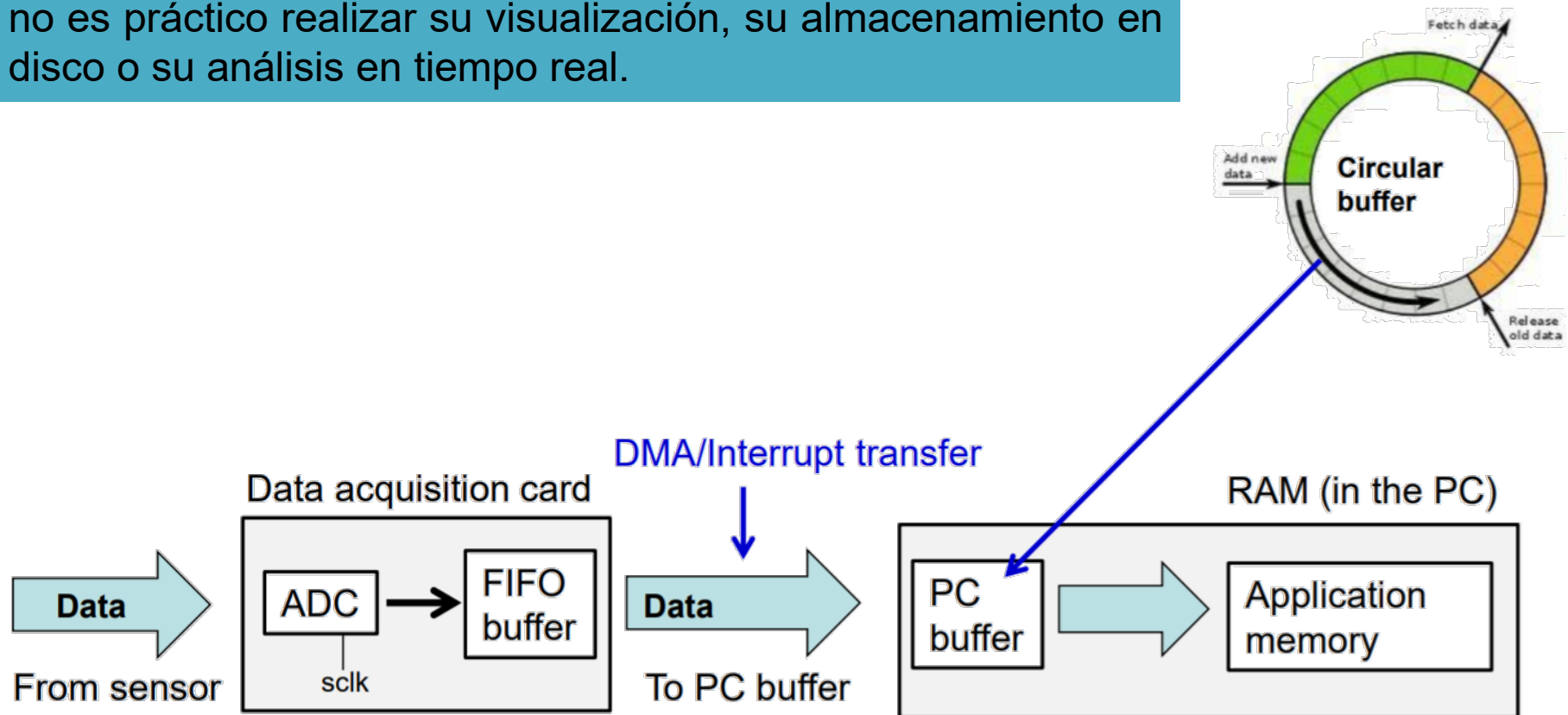
Los datos se transfieren desde el búfer FIFO de la tarjeta DAQ (de tamaño fijo) a la RAM del PC usando interrupciones o DMA, a través de los buses de entrada/salida del PC.

Para implementar una adquisición continua de datos en un sistema que no es de tiempo real se necesita un búfer en el PC. Este búfer se implementa como un **búfer circular** en la memoria RAM del PC.

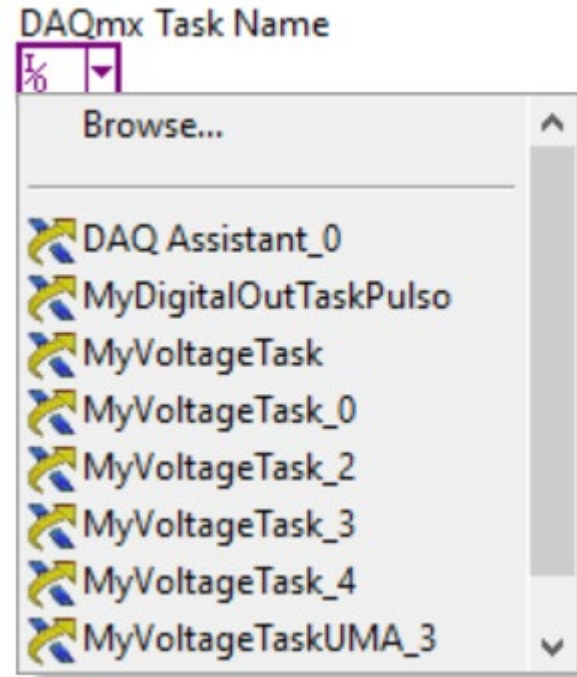
Cuando se realiza una DAQ-read en la aplicación software se leen los valores de este búfer circular en una “variable” de nuestra aplicación.



Se recomienda el uso de un búfer cuando se necesita realizar la adquisición o generación de muestras a una tasa a la que no es práctico realizar su visualización, su almacenamiento en disco o su análisis en tiempo real.

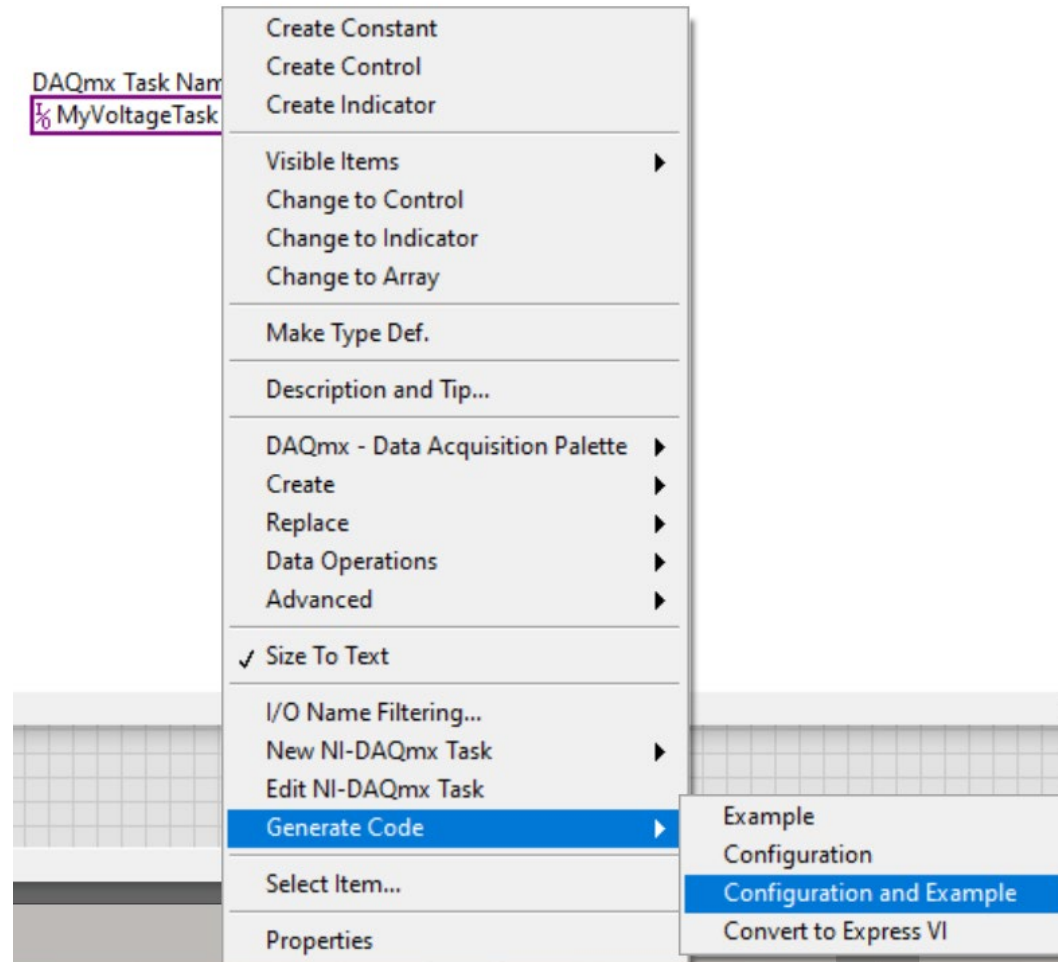


DAQmx Task Name



Usando una constante DAQmx Task Name se puede seleccionar una tarea creada en MAX. También se puede crear una tarea nueva seleccionando “Browse ...”

# Generar configuración y código de ejemplo

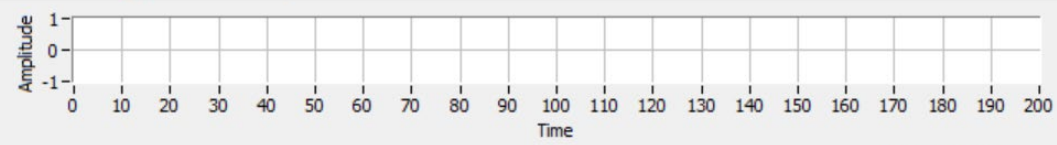


# Generar configuración y código de ejemplo

DAQ Assistant

Undo Redo Run Add Channels Remove Channels

NI-DAQmx Task Connection Diagram



Graph Display Time AutoScale Y-Axis

Configuration Triggering Advanced Timing Logging

Channel Settings

- + X Add Channels Remove Channels Details
- Voltage\_0
- Voltage\_1
- Voltage\_2
- Voltage\_3

*Click the Add Channels button (+) to add more channels to the task.*

Voltage Input Setup

Settings

Signal Input Range

Max 10

Min -10

Scaled Units

Volts

Terminal Configuration

RSE

Custom Scaling

<No Scale>

Timing Settings

Acquisition Mode

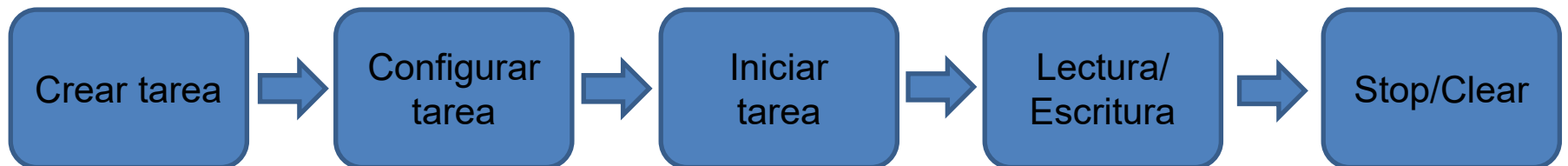
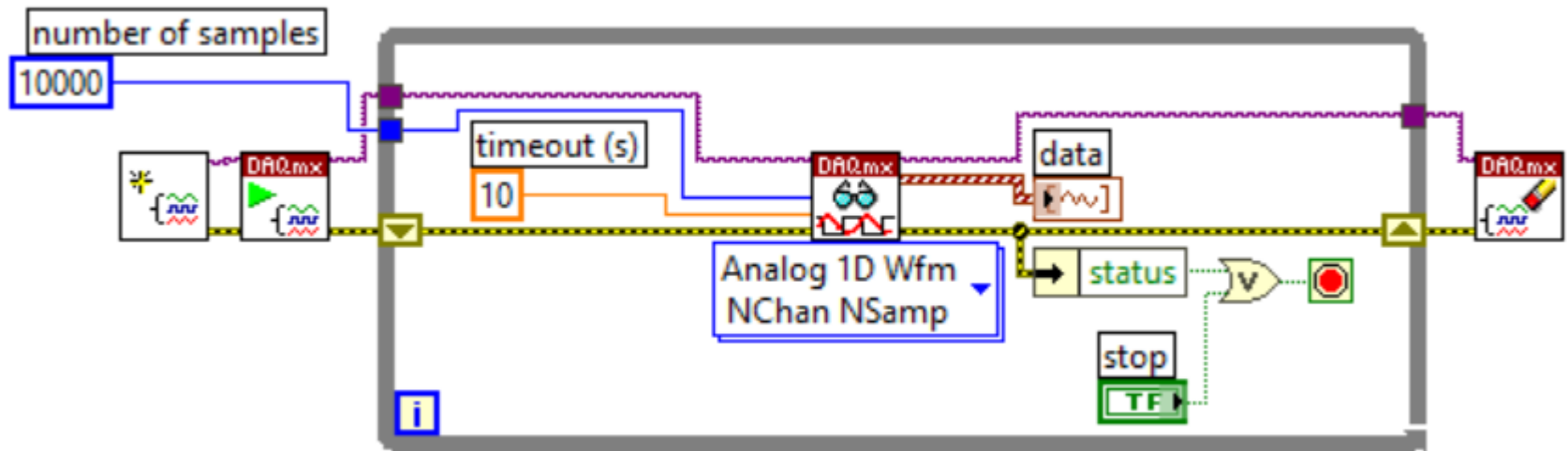
Continuous Samples

Samples to Read

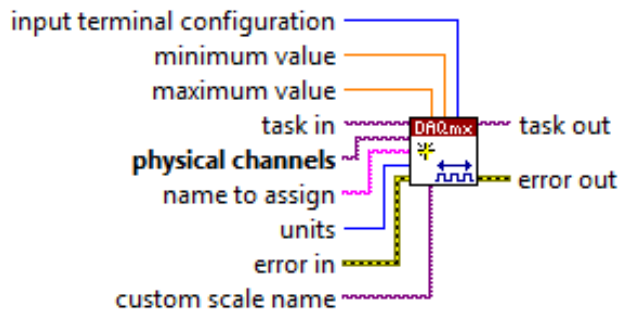
1k

Rate (Hz)

10k



## DAQmx Create Channel (AI-Voltage-Basic).vi



**Creates a virtual channel or set of virtual channels and adds them to a task.** The instances of this [polymorphic VI](#) correspond to the I/O type of the channel, such as analog input, digital output, or counter output; the measurement or generation to perform, such as temperature measurement, voltage generation, or event counting; and in some cases, the sensor to use, such as a thermocouple or RTD for temperature measurements.

If you use this VI without specifying a **task in**, NI-DAQmx creates a new task

Usando la ayuda contextual para este VI, verá que sólo se requiere la entrada del canal físico. “Input terminal configuration” tiene un valor predeterminado según el dispositivo DAQ, pero es importante verificar que se esté utilizando la configuración de terminal correcta para la situación de conexión a tierra que entre el sensor y el dispositivo DAQ.

**1) Crear una tarea en Labview configurando una adquisición de una señal de voltaje del tipo “N muestras”, configurar un número de muestras igual a 10000 y una frecuencia de muestreo de 10kHz. Configurar el timeout correctamente.**

**Generar el código de configuración y ejemplo de uso de la tarea creada.**

**Crear una tarea en Labview configurando una adquisición “continua” con 10000 muestras y una frecuencia de muestreo de 10kHz. Configurar el timeout correctamente.**

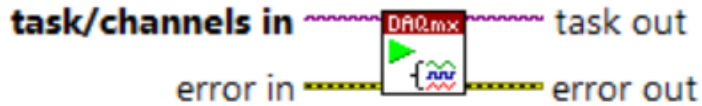
**Generar el código de configuración y ejemplo de uso de la tarea creada.**

**¿Cuáles son las diferencias entre ambos códigos generados? ¿Cuál es el objetivo del shift register?**

**2) Crear una tarea en Labview para la adquisición simultánea de 4 señales de entrada de tipo voltaje, realizar una adquisición continua, configurar 10000 y una frecuencia de 10kHz. Generar el código de configuración y ejemplo de uso de la tarea creada. Ejecutar el código resultante.**

**¿Qué ocurre durante la ejecución?**

## DAQmx Start Task.vi



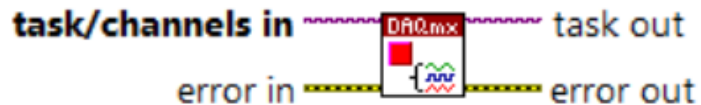
Transitions the task to the running state to begin the measurement or generation. Using this VI is required for some applications and is optional for others.

Iniciar la tarea significa que el dispositivo está listo para tomar una medida (DAQmax Start Task incluye un **trigger implícito**).

Si se usan las funciones DAQmx Write and Read dentro de un bucle el uso de Start Task y Stop Task mejora significativamente el rendimiento.

Cualquier configuración de la tarea se debe hacer antes de llamar a DAQ Start Task.vi

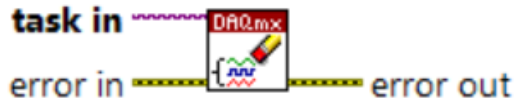
## DAQmx Stop Task.vi



Stops the task and returns it to the state the task was in before the DAQmx Start Task VI ran or the DAQmx Write VI ran with the **autostart** input set to TRUE.

**DAQmx Stop Task VI**, simplemente detiene la tarea y no libera los recursos. Puede volver a iniciar la tarea sin volver a crearla.

## DAQmx Clear Task.vi



Clears the task. Before clearing, this VI aborts the task, if necessary, and releases any resources the task reserved. You cannot use a task after you clear it unless you recreate the task.

**DAQmx Clear Task VI**, detiene la tarea y la borra. Antes de borrar la tarea libera cualquier recurso que la tarea haya reservado. Debe volver a crear la tarea antes de poder volver a utilizarla.

Normalmente se usan ambos aunque el DAQmx Clear task VI parará la tarea si es necesario.

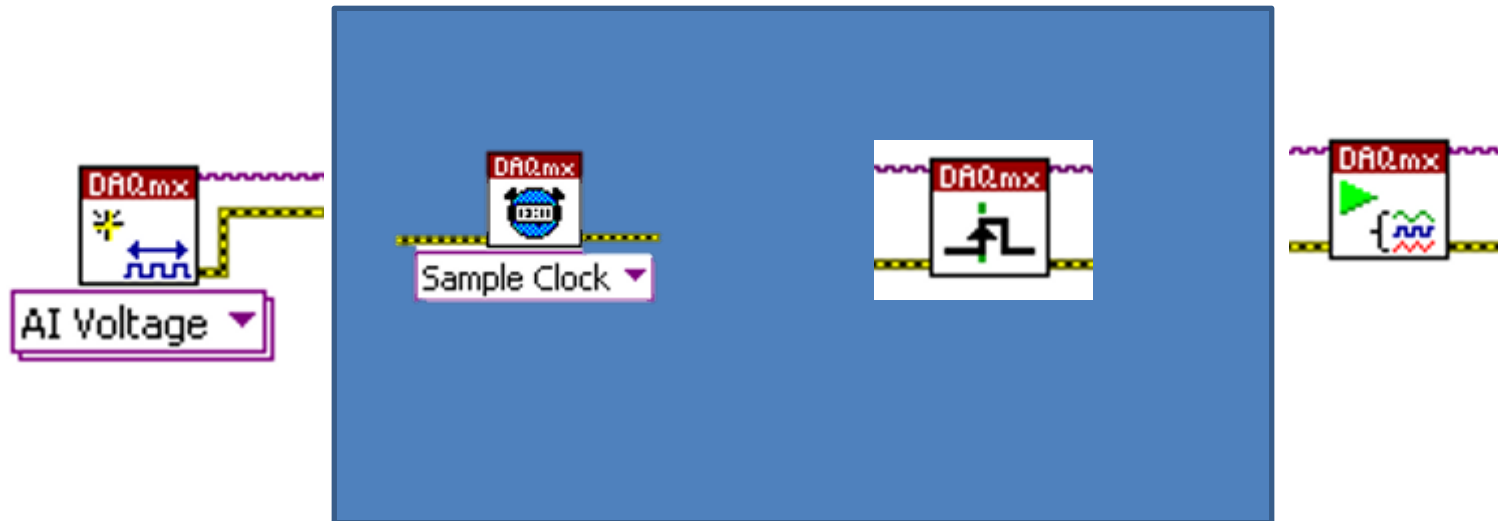
Si se planea realizar la misma adquisición varias veces en el programa, llamar al **DAQmx Clear Task VI** sería ineficiente en comparación con el **DAQmx Stop Task VI**. Con los mismos parámetros físicos y de tiempo, el método preferido es iniciar y detener la misma tarea una y otra vez.

Cuando un dispositivo controlado por NI-DAQmx hace algo se dice que está realizando una acción. Dos acciones muy comunes son producir una muestra e iniciar una adquisición de forma de onda. Cada acción NI-DAQmx necesita un estímulo o causa. Cuando se produce el estímulo, se realiza la acción. Las causas de las acciones se llaman **disparos**. Los **disparos** se nombran después de las acciones que causan, como un disparo de inicio para iniciar una adquisición.



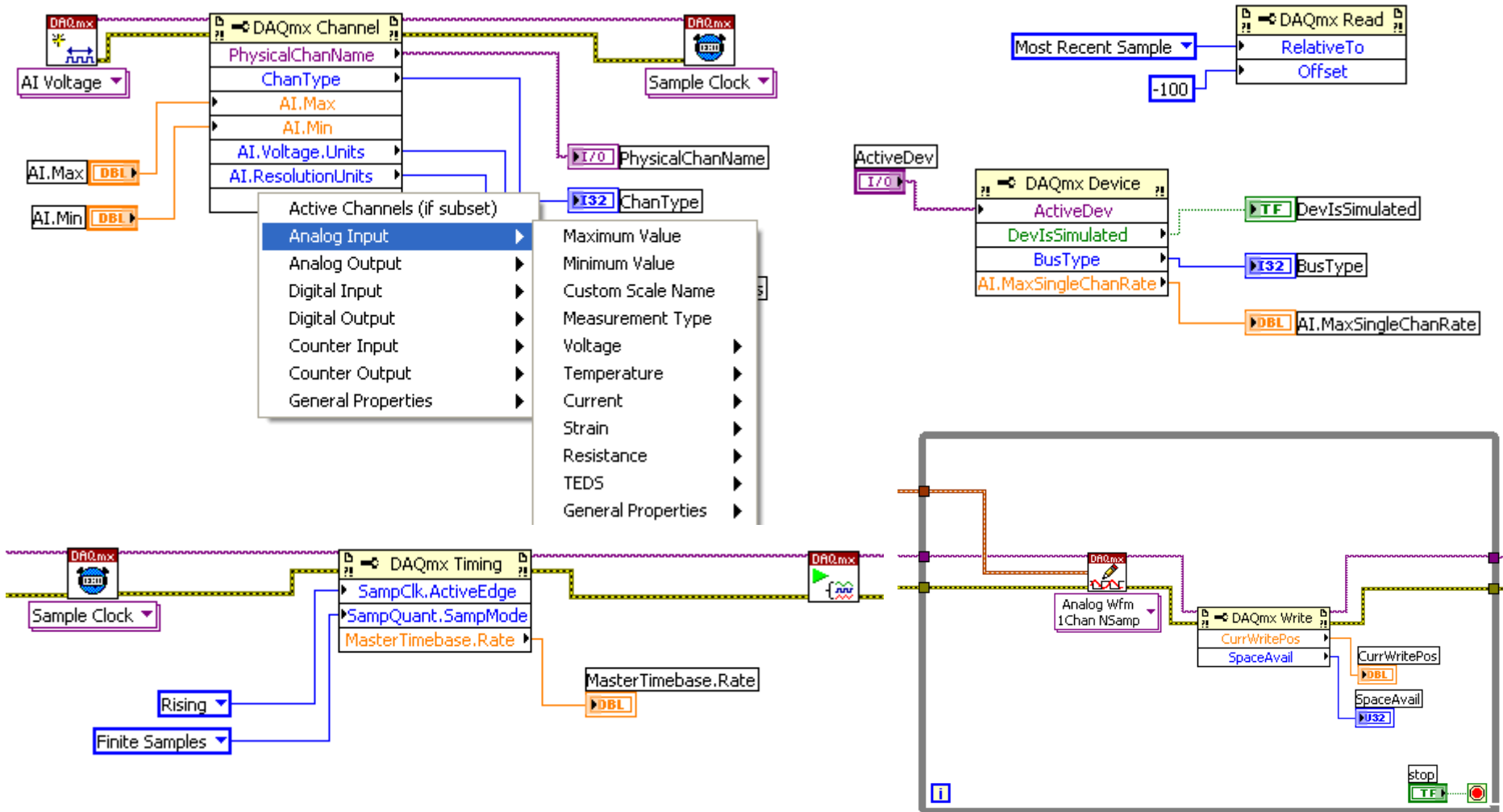
La función NI-DAQmx Trigger configura un disparo para realizar/iniciar una acción. Las acciones más usadas son un **disparo de inicio** y un **disparo de referencia**. Un disparo de inicio inicia una adquisición o generación. Un disparo de referencia establece la ubicación, en un juego de muestras adquiridas, dónde terminan los datos de pre-disparo y comienzan los datos de post-disparo. Estos dos disparos pueden ser configurados para ocurrir en un borde digital, un borde analógico o cuando una señal analógica entra o deja una ventana.

Cualquier configuración de la tarea debe realizarse antes de la llamada a **DAQmx Start task VI**.



Los nodos de propiedades de NI-DAQmx proporcionan acceso a todas las propiedades asociadas a la operación de adquisición de datos

Los nodos de propiedades de NI-DAQmx proporcionan acceso a todas las propiedades asociadas a la operación de adquisición de datos



## [polymorphic VI](#)

Propio de National

Habituales otras APIS

**Waveform** —The waveform data format includes the channel name, timing, and unit information with the actual 64-bit scaled floating-point data. Because there is overhead associated with including this additional information, NI-DAQmx allows you to configure the information you want to include.

**64-Bit Floating-Point Numbers**—The 64-bit floating-point number format allows you to read or write scaled data with no additional information. Use this format to work with scaled data that requires higher performance than the waveform format provides. You might also use this format **because it is a better match for the libraries you plan to use.**

**Unsigned and Signed Integers** —The unsigned and signed integer format reads or writes data in the native format of the device. Use this format for maximum performance. The tradeoff is that your application has to understand how to interpret and manipulate data that is not in engineering units.

Doubles (ya escalados)

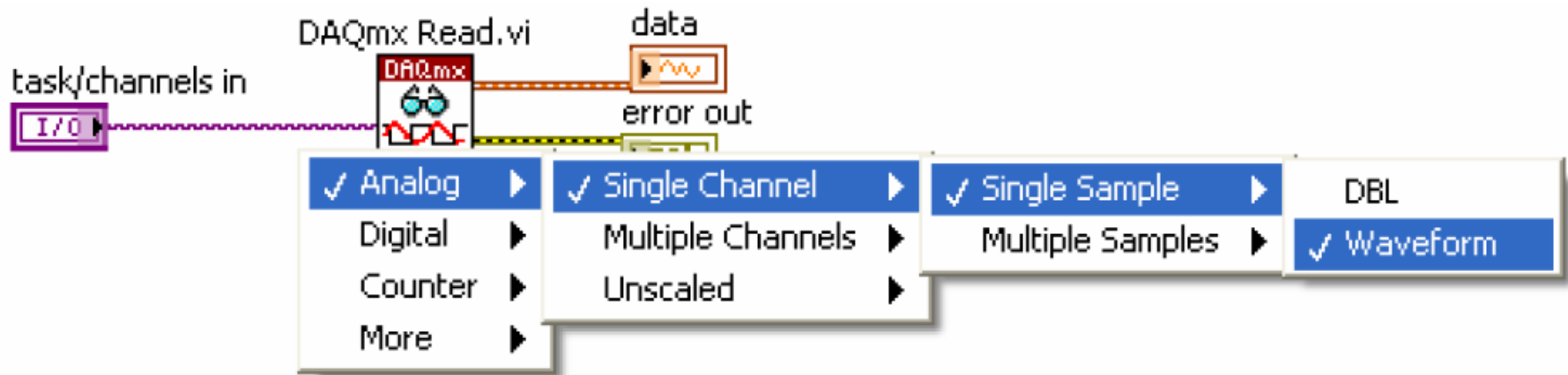
Integer (Raw)

### DAQmx Read (Analog 1D Wfm NChan 1Samp).vi



Reads a waveform that contains a single sample from each channel in a task that contains one or more analog input channels.

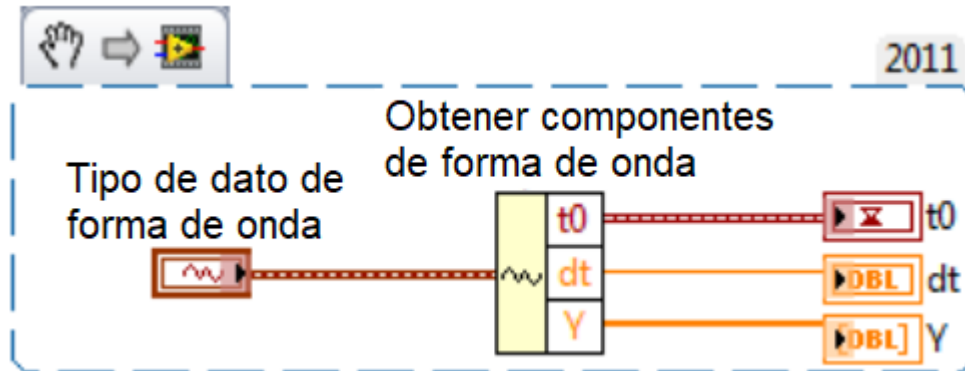
[Detailed help](#)



En el primer menú se elige el tipo de entrada. En el segundo menú se determina el número de canales o si los datos están sin escalar. En el tercero se selecciona el número de muestras a leer. En el cuarto menú el modo que se van a enviar los datos.

El tipo de dato **waveform** es usado por LabVIEW para mostrar y almacenar mediciones de señal periódicas.

El tipo de dato de **waveform** es un grupo compuesto por los siguientes componentes:



## Componente Descripción

t0	Una marca de tiempo que representa el tiempo en que comienza la señal.
dt	Un doble numérico que describe la diferencia en el tiempo en segundos entre cada muestra en la señal.
Y	Un arreglo de 1D de dobles que contiene los valores de las muestras de esa señal.

## Adquisición de 1 muestra

La adquisición de una sola muestra es una operación bajo demanda. En otras palabras, el controlador adquiere un valor de un canal de entrada y devuelve inmediatamente el valor. Esta operación no requiere de almacenamiento temporal ni temporización hardware.

## Adquisición finita de N muestras

- **Punto a punto.** La transferencia de los datos adquiridos es inmediata. Los requisitos de entrada/salida son más exigentes de cara al bus utilizado.
- **Adquisición de forma de onda.** Se recomienda usar esta adquisición que usa un búfer en memoria y que permite, por tanto, adquirir datos con más eficacia. Se debe especificar la velocidad de muestreo y el modo de muestreo (finito) en la función **DAQmx timing VI**.

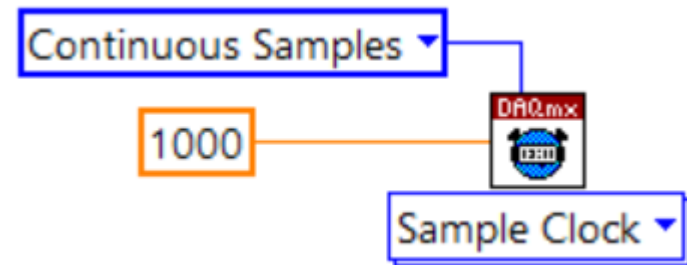
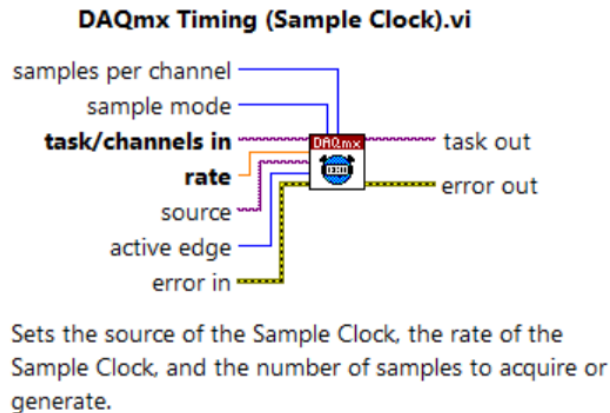
## Adquisición continua

Si se necesita ver, procesar o registrar un subconjunto de muestras mientras se adquieren, se debe adquirir muestras continuamente. Para este caso de uso de debe configurar el muestreo continuo.

Cuando se usa/configura el “**DAQmx Timing.vi**” DAQmx configura la placa para llevar acabo una E/S temporizada por **hardware**.

- Se puede usar el reloj interno de la tarjeta o un reloj externo

Al habilitar el **muestreo continuo** DAQmx configura automáticamente un **búfer circular en RAM**.



**Número de muestras por canal:** especifica el número de muestras a adquirir o generar por cada canal en la tarea si el modo es el “N samples”.

Si el modo es “Continuo”, NI-DAQmx usa este valor para determinar el tamaño del búfer.

La reserva de memoria es manejada de forma automática si se utiliza el VI de DAQmx Timing:

Si la adquisición es finita (*sample mode* en el DAQmx Timing.vi se configura como **Finite Samples**), NI-DAQmx reserva un espacio en memoria para el búfer igual al tamaño especificado en el valor de *samples per channel*.

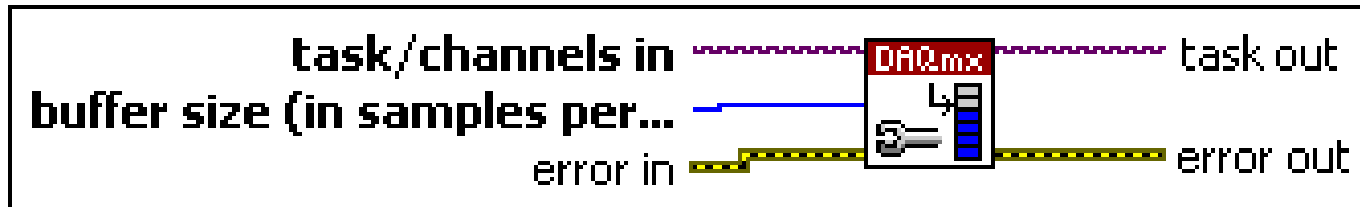
Si la adquisición es continua (modo de muestreo en la función de Timing se configura como Continuous Samples), NI-DAQmx reserva un búfer de igual tamaño que el número de muestras por canal configurado en el atributo o propiedad, a menos que este valor sea menor que los valores listados en la siguiente tabla.

Tasa de muestreo	Tamaño de búfer
0 - 100 S/s	1 kS
100 - 10,000 S/s	10 kS
10,000 - 1,000,000 S/s	100 kS
> 1,000,000 S/s	1 MS

# ¿Cómo se modifica el tamaño del búfer creado por DAQmx?

Es posible sobrescribir manualmente la opción de DAQmx del tamaño del buffer al llamar al **VI DAQmx Configure Input Buffer** (o DAQmx Configure Output Buffer.vi para operaciones de salida) que se encuentra en la paleta de **NI Measurements » DAQmx - Data Acquisition » DAQmx Advanced Task Options**.

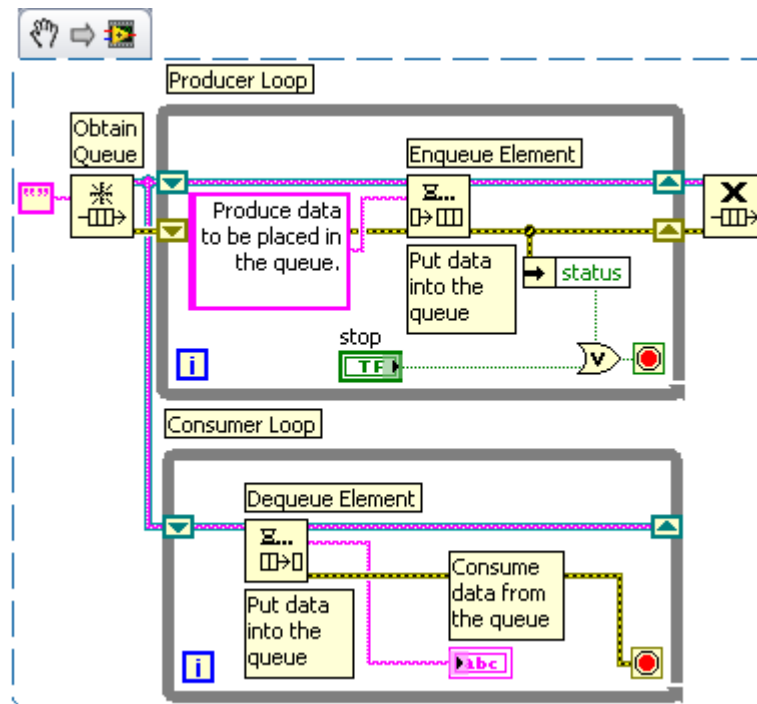
Este VI sobrescribe la reserva de espacio en memoria que DAQmx realiza automáticamente.



Un **error de sobrescritura** indica que la información se pierde y ocurre cuando el programa software no lee datos del búfer del PC lo suficientemente rápido. Las muestras escritas en el búfer circular son sobrescritas antes de ser leídas.

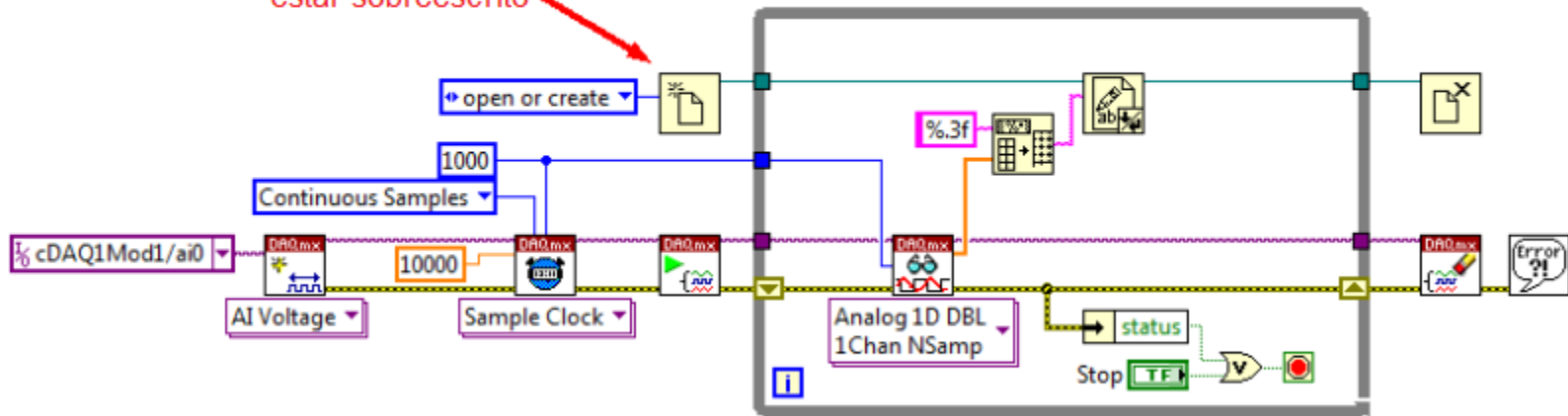
## Error -200279

Solución: Aumentar el tamaño del búfer, leer más rápido o escribir más lentamente, patrón de diseño Productor/Consumidor

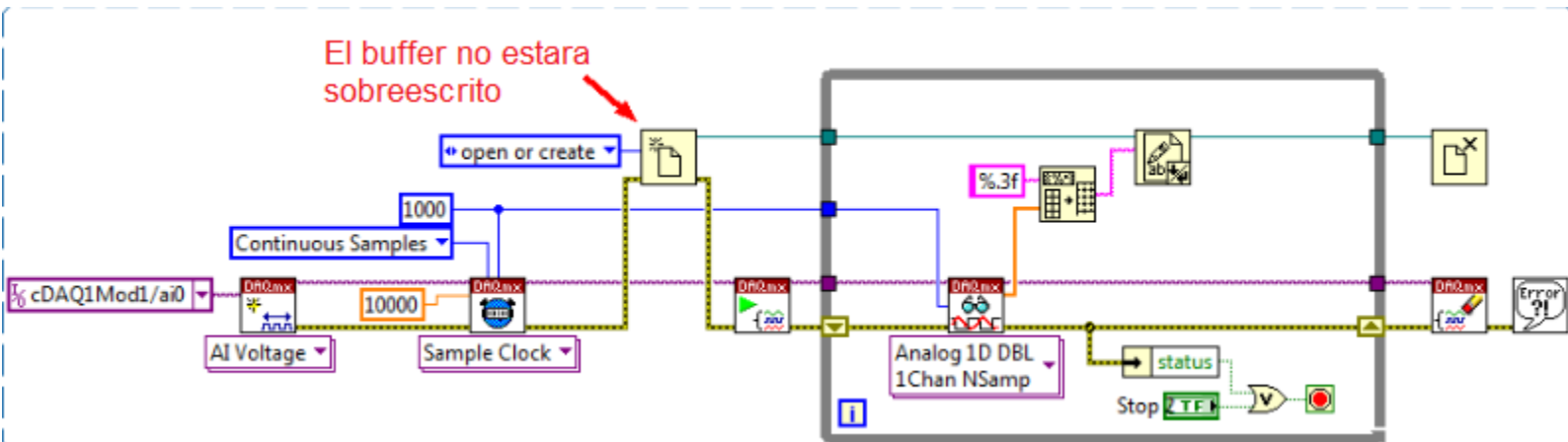


# Error de sobrescritura

El buffer puede estar sobrescrito



El buffer no estara sobrescrito



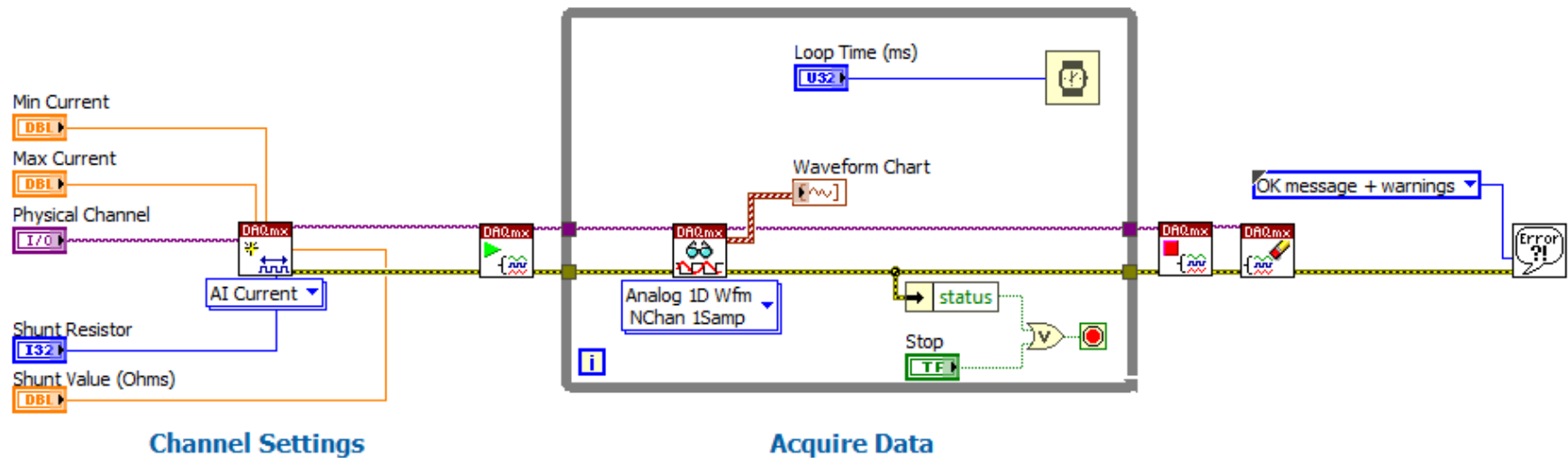
Un **error de desbordamiento** indica que la información se ha perdido antes, en el proceso de adquisición de datos. Los errores de desbordamiento indican que el búfer FIFO de la tarjeta de adquisición de datos ha alcanzado su capacidad máxima para almacenar las muestras adquiridas y ya no puede aceptar nuevas muestras. Un error de desbordamiento es sintomático de una velocidad de transferencia de bus que no alcanza la velocidad de entrada de datos solicitada.

## **Error -200361**

Solución: utilizar un mecanismo de transferencia de acceso directo a memoria (DMA), disminuir la velocidad de entrada de datos solicitada o reducir la cantidad de dispositivos que comparten el bus PCI

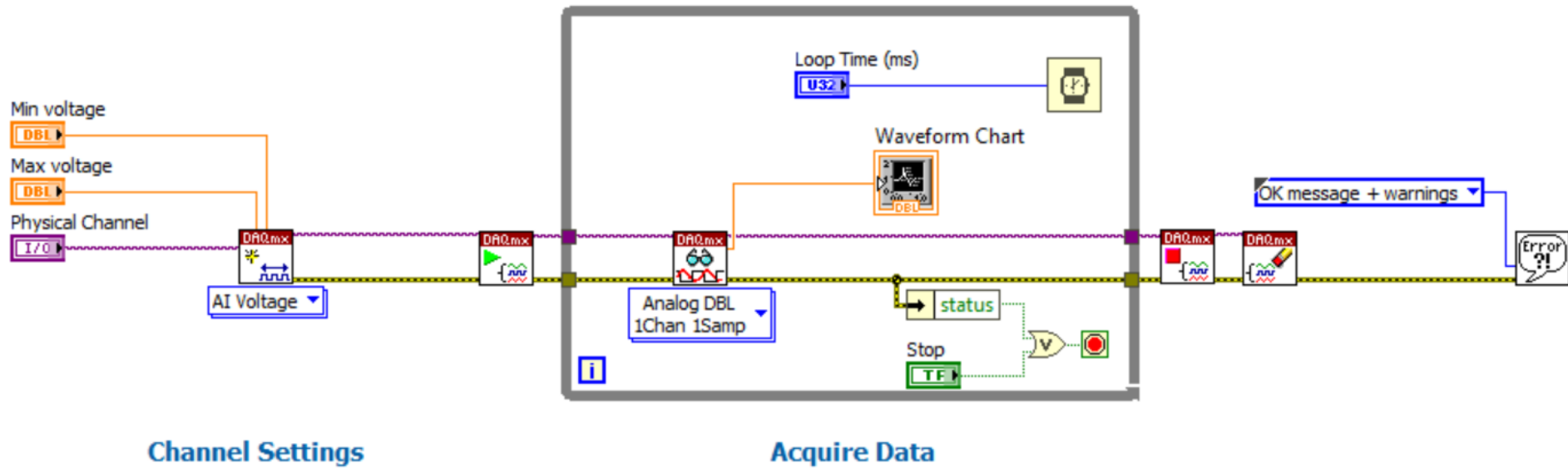
1. Muestreo de un número finito de muestras punto a punto con un temporización software
2. Muestreo de un número finito de muestras usando un búfer con temporización hardware
3. Muestreo continuo con búfer
4. Muestreo continuo con búfer e inicio de captura dirigido por disparo

# Ejemplo 1: Muestreo de un número finito de muestras punto a punto con un temporización software



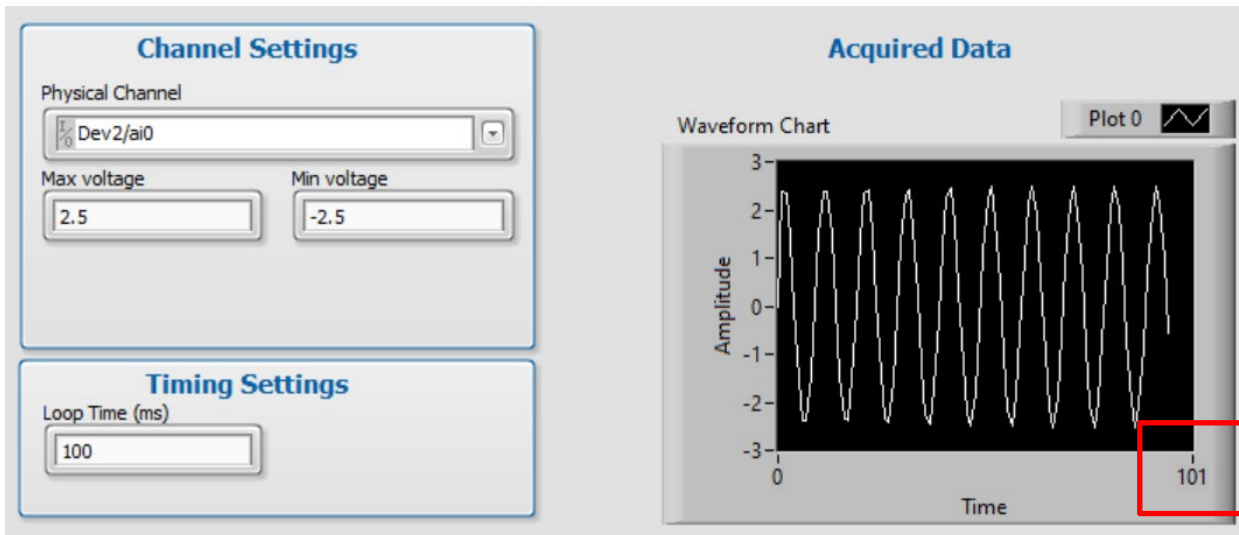
Se ha configurado la adquisición de un solo punto en el DAQmx Read VI. Para la adquisición de múltiples muestras se usa un bucle for con una función de temporización que permite especificar el número de muestras y la tasa de lectura.

# Adquisición software punto a punto



Channel Settings

Acquire Data

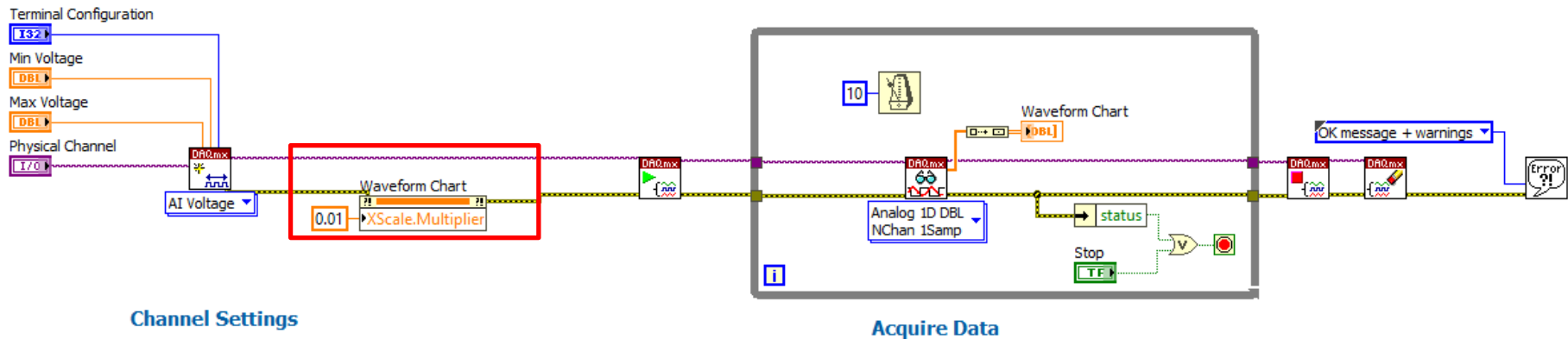


La señal muestreada tiene un frecuencia de 1 Hz.

El eje temporal no está escalado respecto al tiempo.

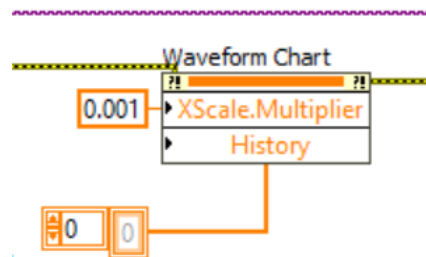
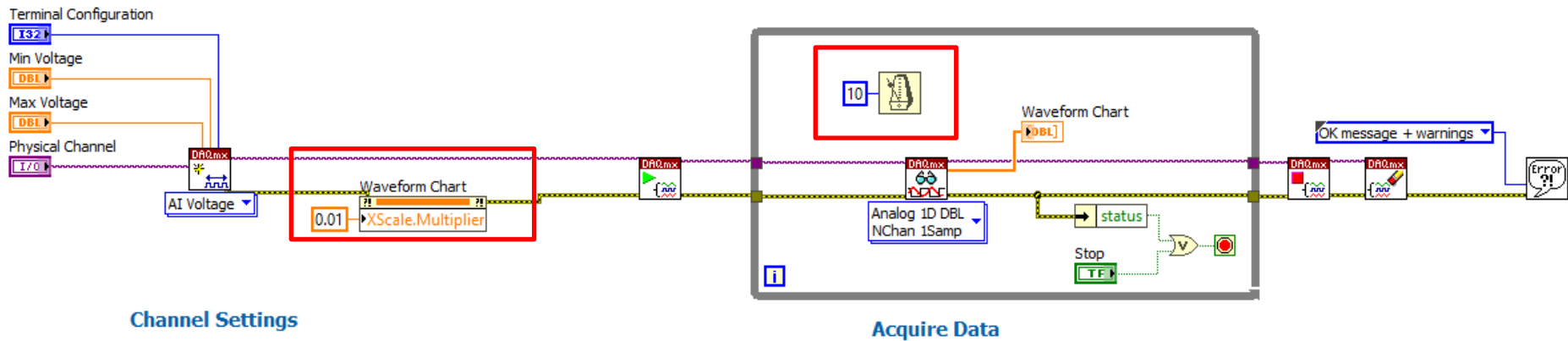
# Modificar las unidades del Eje X en un Gráfico a Segundos

Trazar una matriz de datos en lugar de una forma de onda. En la matriz de datos el eje X se escala por el inverso de su frecuencia de muestreo



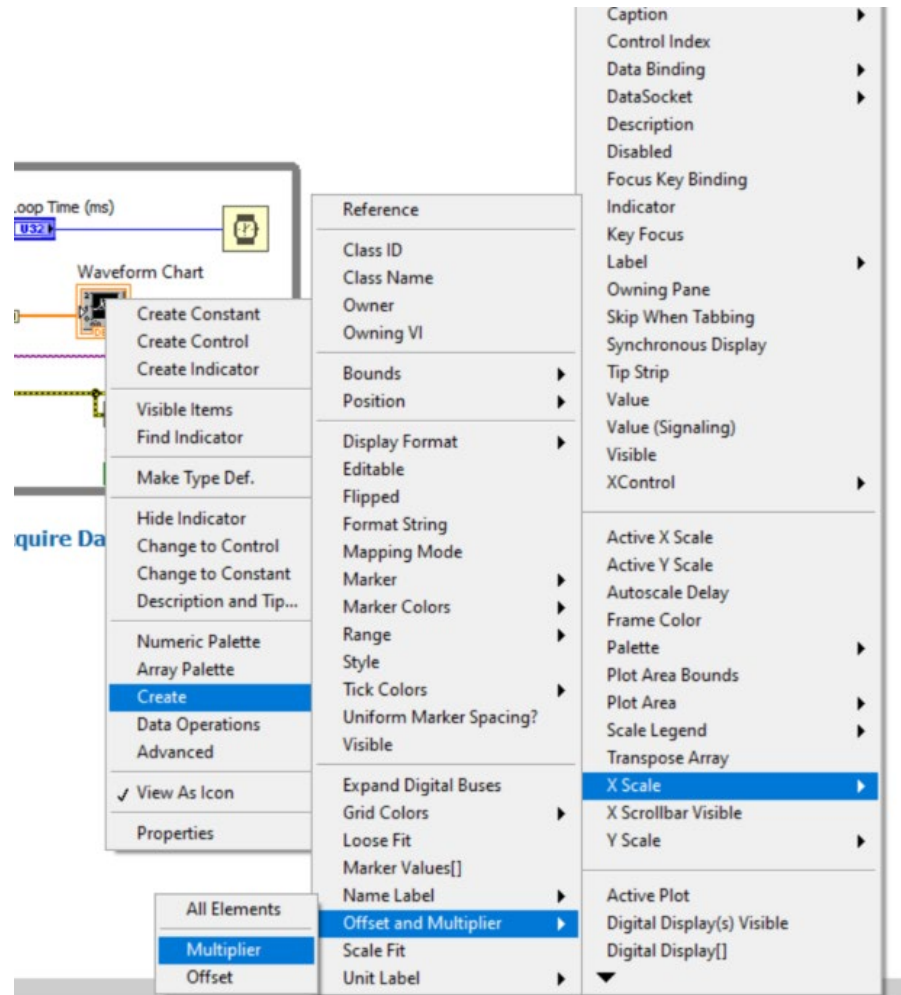
# Modificar las unidades del Eje X en un Gráfico a Segundos

Trazar una matriz de datos en lugar de una forma de onda. En la matriz de datos el eje X se escala por el inverso de su frecuencia de muestreo



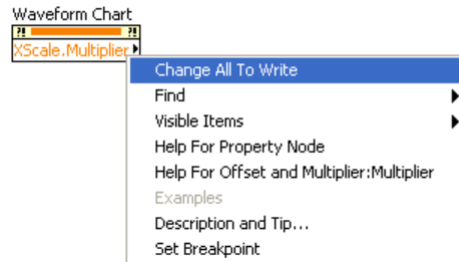
# Modificar las unidades del Eje X en un Gráfico a Segundos

Creación un nodo de propiedad asociado al gráfico: Create – Property Node – X Scale – Offset and Multiplier - Multiplier

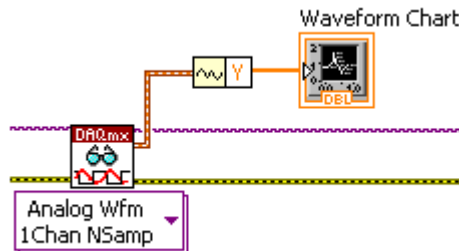


# Modificar las unidades del Eje X en un Gráfico a Segundos

Seleccionar la opción “Change All to Write” en el nodo de propiedad creado.



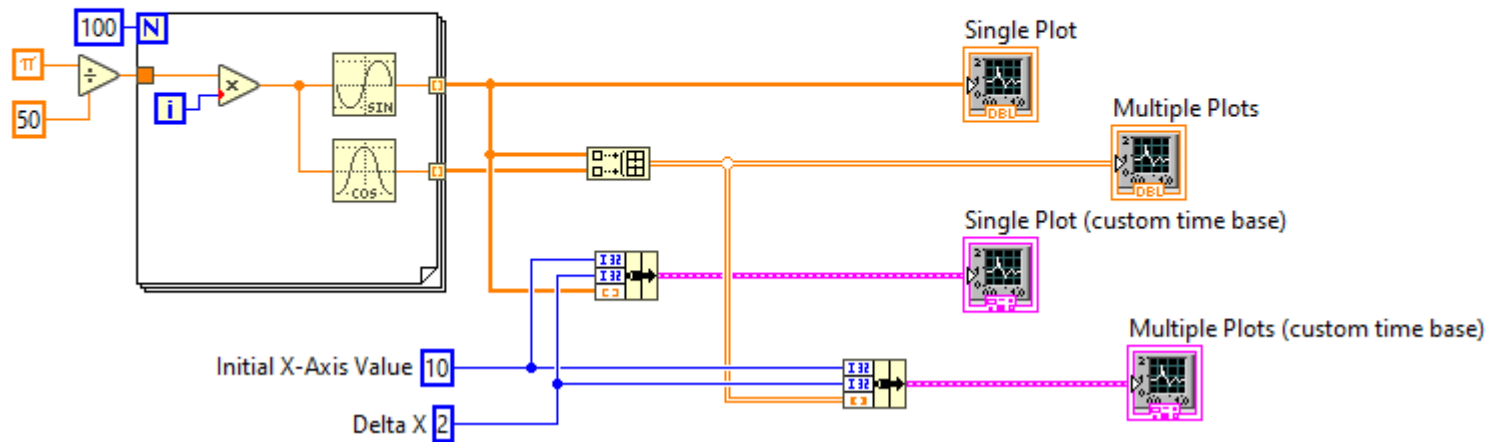
Si se usa el formato forma de onda extraer la componente “y”.



En el caso de usar la matriz DAQmx Read 1D o 2D, trazar directamente la matriz resultante.

# Modificación valor inicial y delta del eje X

The graphs that use the default display behavior of the Waveform Graph are array data types. The graphs that define custom X-Axis initial values and delta values are cluster data types.



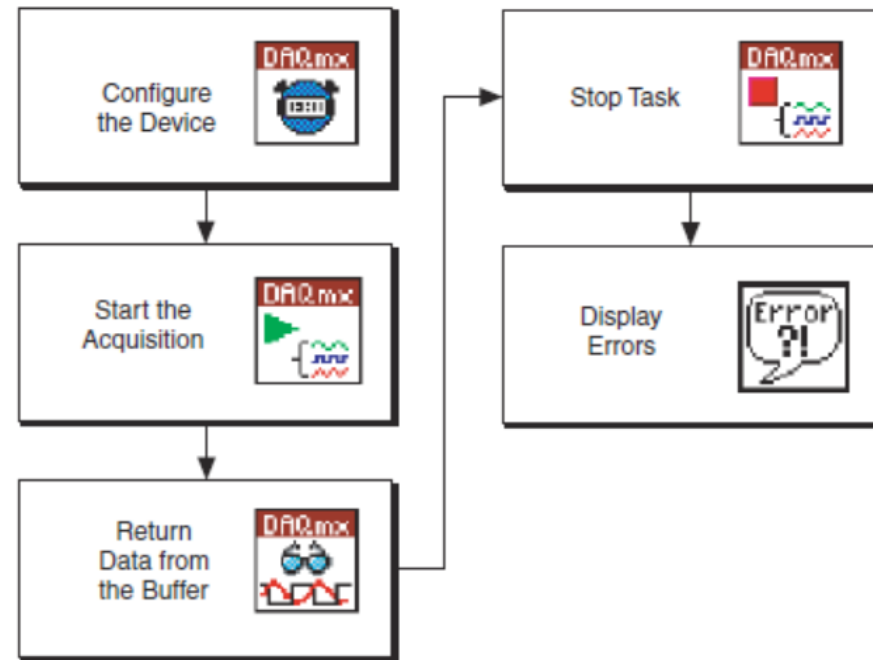
# Ejemplo 2: Muestreo de un número finito de muestras usando un búfer con temporización hardware

El siguiente diagrama de flujo muestra una adquisición de muestras con búfer básica en la que se adquiere un número de puntos especificados a la tasa indicada.

Se utiliza el **DAQmx Timing VI** para configurar la sincronización y el tamaño del búfer.

A continuación se utiliza **DAQmx Start Task VI** para iniciar la adquisición.

El **DAQmx Read VI** espera hasta que todas las muestras en cada canal estén disponibles antes de devolver el datos y seguir adelante.

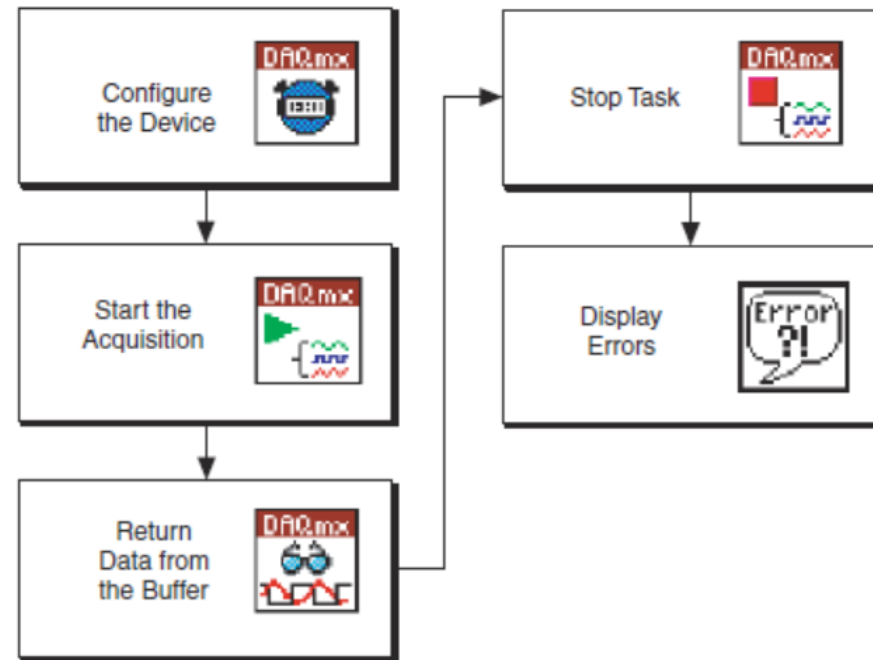


# Ejemplo 2: Muestreo de un número finito de muestras usando un búfer con temporización hardware

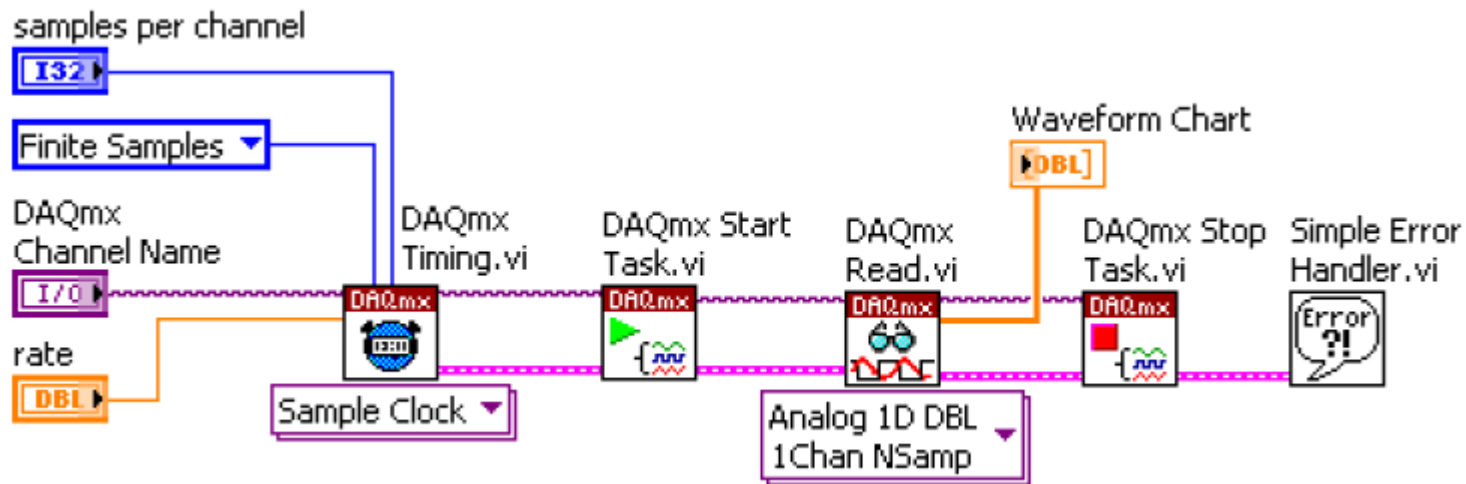
**DAQmx Read** tiene una entrada llamada **Number of Samples per Channel**. El valor de **Number of Samples per Channel** determina el tamaño, en muestras, de cada matriz o forma de onda de datos que se leen desde el buffer a su aplicación cuando se ejecuta **DAQmx Read**.

El valor predeterminado para **Number of Samples per Channel** es **-1**, lo que hace que **DAQmx Read** lea todos los datos que están disponibles en el buffer inmediatamente.

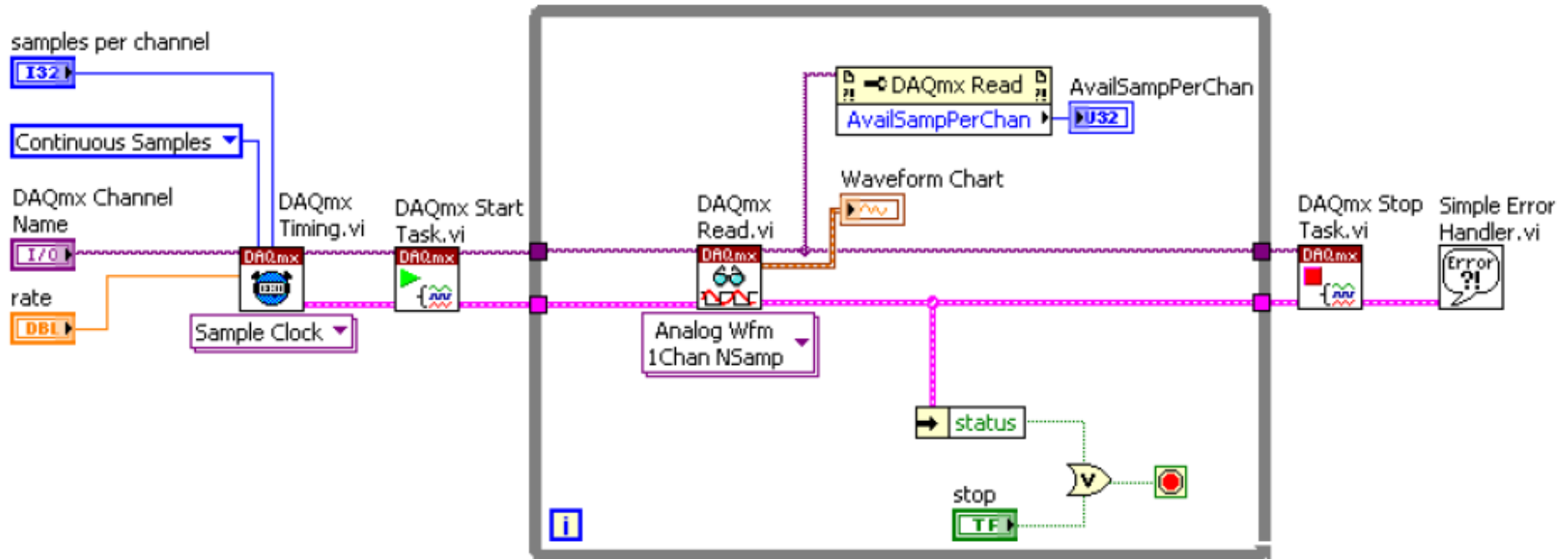
**DAQmx Stop Task VI** detiene la tarea y libera los recursos asignados al dispositivo. El controlador de errores VI muestra cualquier error ocurrido durante el proceso.



# Ejemplo 2: Muestreo de un número finito de muestras usando un búfer con temporización hardware



# Ejemplo 3: Muestreo continuo con búfer

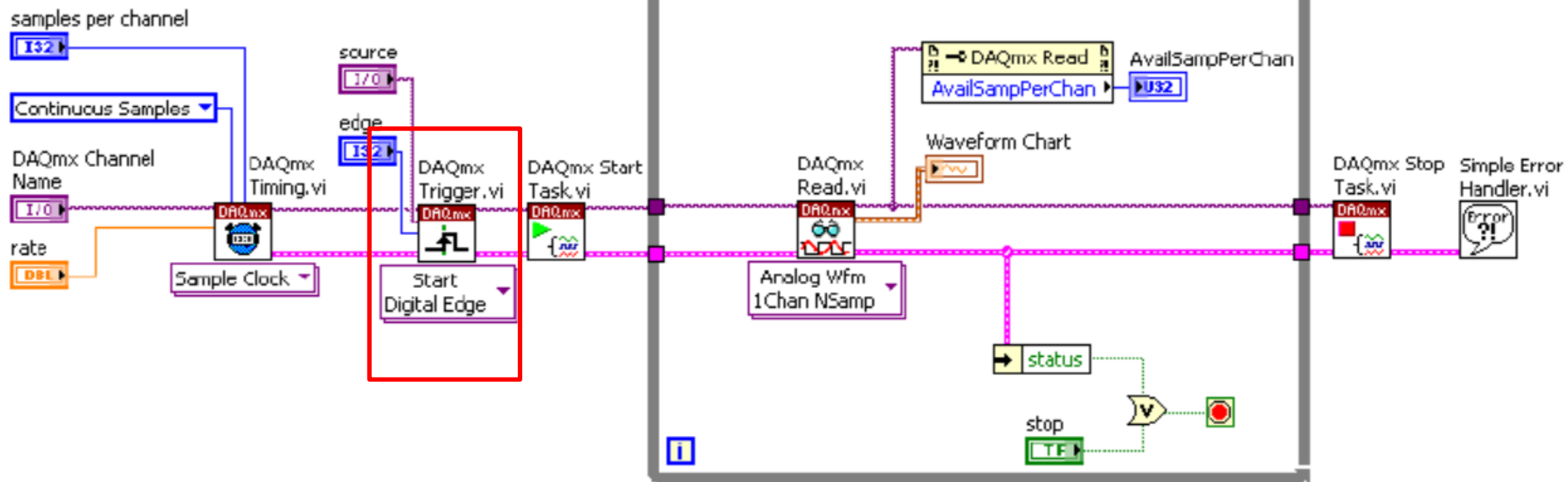


Para evitar el desbordamiento del búfer, el número de muestras por canal para leer no puede ser igual al tamaño del búfer porque la función **DAQmx Read** no se completará hasta que se lean todas las muestras que espera recibir. Es una buena práctica configurar el número de muestras por canal en la función DAQmx Read a un cuarto o la mitad del tamaño del búfer para una adquisición continua con búfer. LabVIEW envía continuamente datos al búfer, es importante controlar el número de muestras disponibles en el búfer para asegurarse de que el búfer esté vaciando lo suficientemente rápido.

## Ejemplo 3: Muestreo continuo con búfer. Error de sobreescritura

- **Aumentar el número de muestras por canal (tamaño del búfer) con el DAQmx Timing VI.** El aumento del tamaño del búfer no resuelve el problema si no se vacía el búfer lo suficientemente rápido (recuerda el configurar las muestras por canal para leer a un cuarto o la mitad del tamaño del búfer).
- **Vaciado del búfer más rápidamente aumentando el número de muestras por canal para leer.** No se debe establecer el número de muestras por canal para leer demasiado alto porque esperará en el DAQmx Read VI el número de muestras por canal en el búfer para igualar el número de muestras por canal para leer. El tiempo dedicado a esperar que las muestras llenen el búfer podría usarse para vaciar.
- **Disminuir la frecuencia de muestreo por canal con DAQmx Timing VI.** Esta configuración ralentiza la velocidad de envío de datos al búfer, pero podría no ser una opción si desea una determinada frecuencia de muestreo.
- **Evitar ralentizar el bucle de lectura con análisis innecesarios.** Usar una arquitectura productor – consumidor.

# Ejemplo 4: Muestreo continuo con búfer e inicio de captura dirigido por disparo



La **compartición de la señal de reloj** es el método más sencillo para sincronizar las medidas y asegurar que los relojes de medición nos se desvíen con el tiempo.

Cuando se comparte un mismo reloj, el dispositivo que proporciona la señal se denomina dispositivo **maestro**. El dispositivo que recibe la señal se denomina dispositivo **esclavo**. El uso de una arquitectura maestro esclavo permite que todos los dispositivos detecten los cambios de temporización en los mismos flacos de reloj.

Para asegurarse de que todos los dispositivos inician sus tareas al mismo tiempo **se debe asegurar que el dispositivo esclavo se configure y se ejecute antes de que el dispositivo máster inicie su actividad**. Existen dos formas de asegurar que esto ocurre:

- Hacer un **merge** de la salida de error del start de la tarea del dispositivo esclavo con la entrada de error del start de la tarea del dispositivo maestro.
- Usar una **estructura flat sequence** e iniciar primero el start en el esclavo y en segundo lugar el start del maestro.

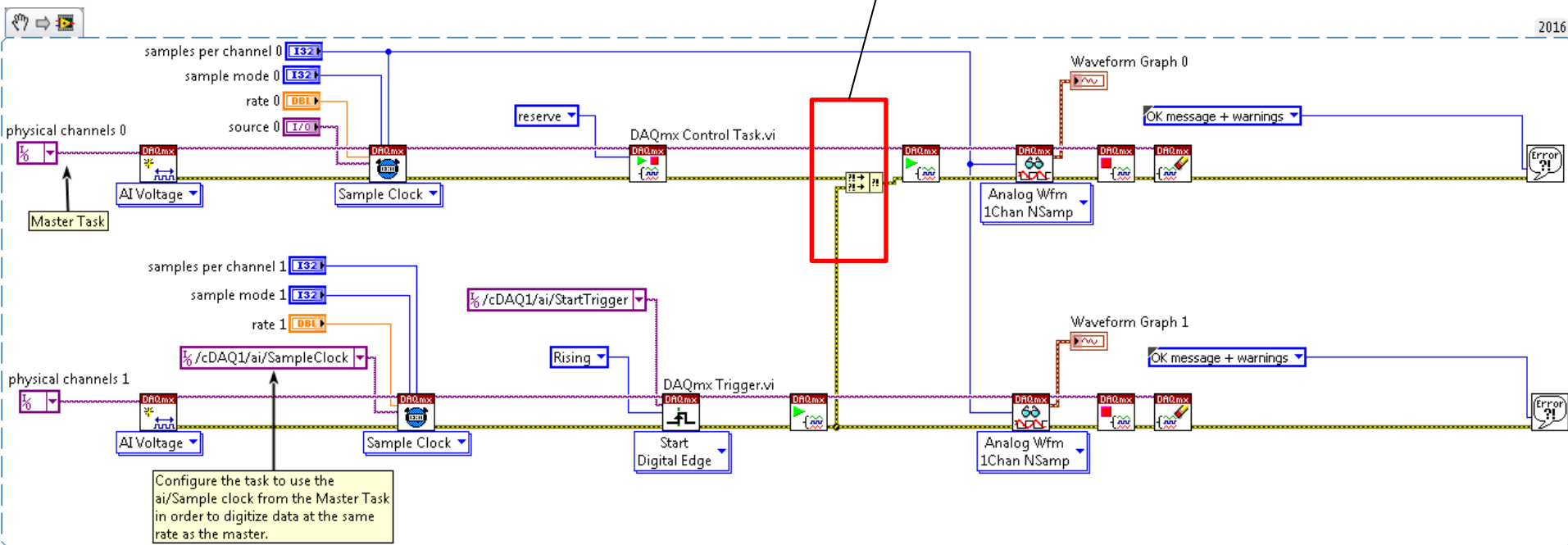
Context Help

### Merge Errors

Merges error I/O clusters from different VIs and functions.

[Detailed help](#)

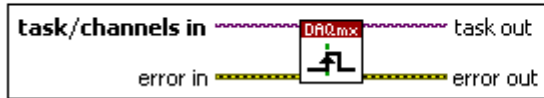
Merge errors



## ■ La tareas del NI-Daq: MÁS Funciones básicas: **Trigger para arrancar adquisición**

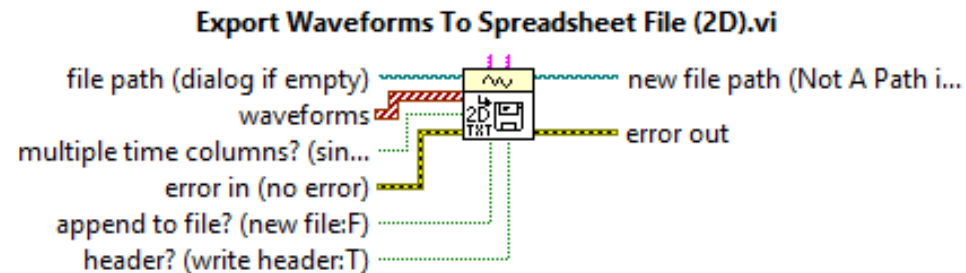
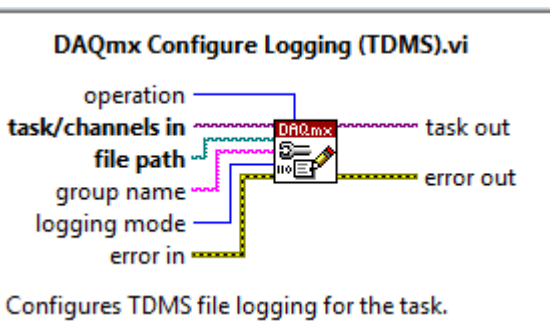
### Start None

Configures the task to start acquiring or generating samples immediately upon starting the task.

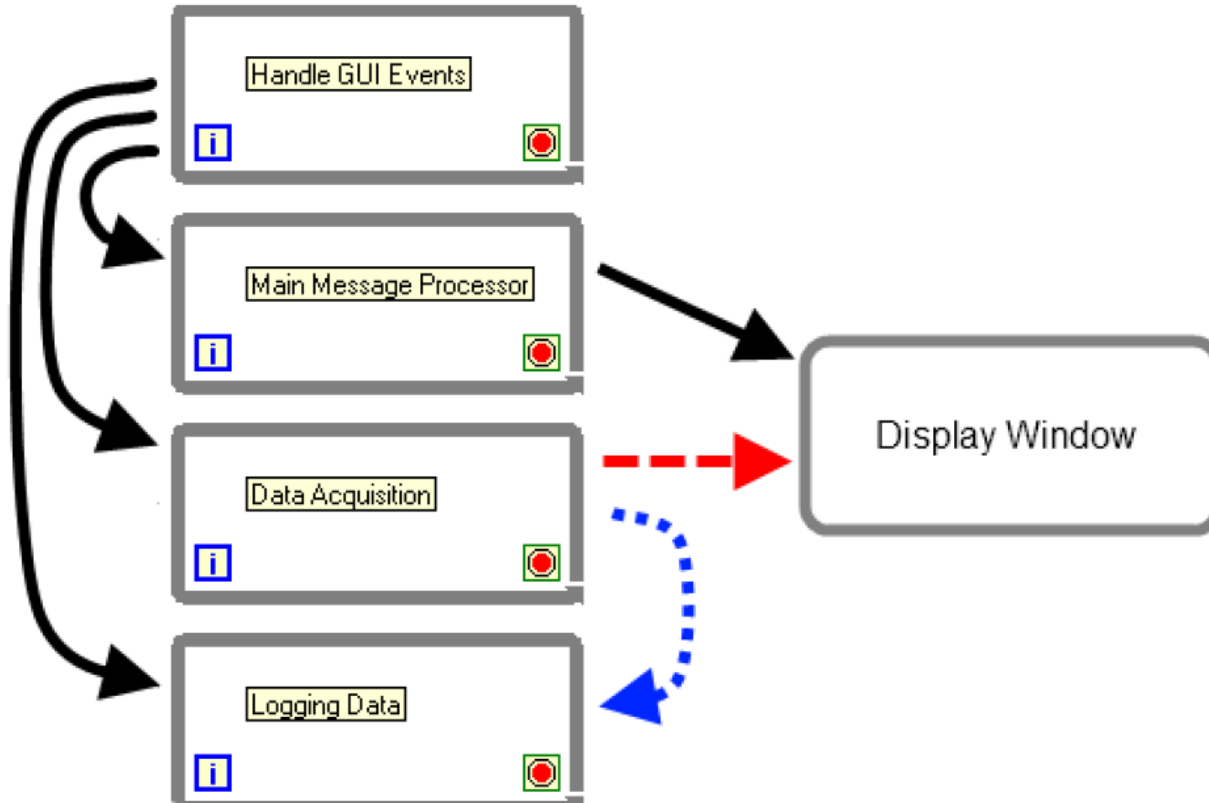


**Trigger explícito.** Un **trigger de inicio** comienza la adquisición de datos  
 El **trigger de referencia** establece el punto de referencia en un conjunto de muestras de entradas. Los datos adquiridos hasta el punto de referencia son datos de pretrigger. Los datos adquiridos después del punto de referencia son datos del posttrigger.

## ■ La tareas del NI-Daq: Funciones extras de Labview: **Almacenamiento de datos**



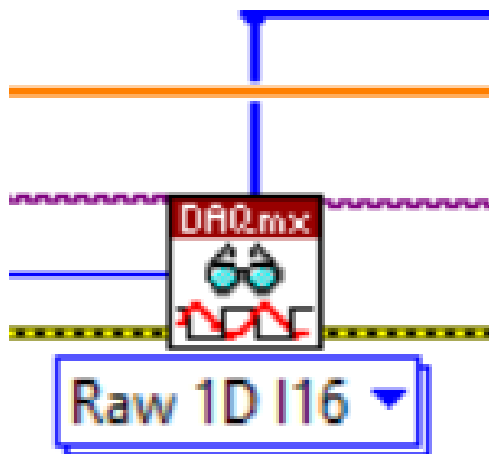
# DAQ avanzado con múltiples bucles while



Un programa DAQ generalmente tiene varios bucles while que se ejecutan en paralelo, y los datos (y mensajes) deben distribuirse entre los bucles utilizando colas.

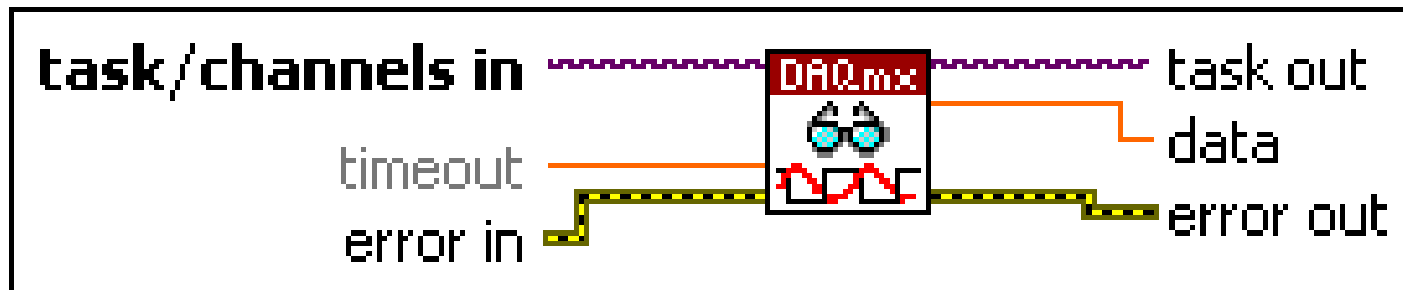
- Black : Messages transfer (using queue)
- Blue : Data transfer using **Queue** (NB: queues have memory – no data is lost)
- Red : Data transfer using **Notifier** (NB: notifiers do not have memory/FIFO)

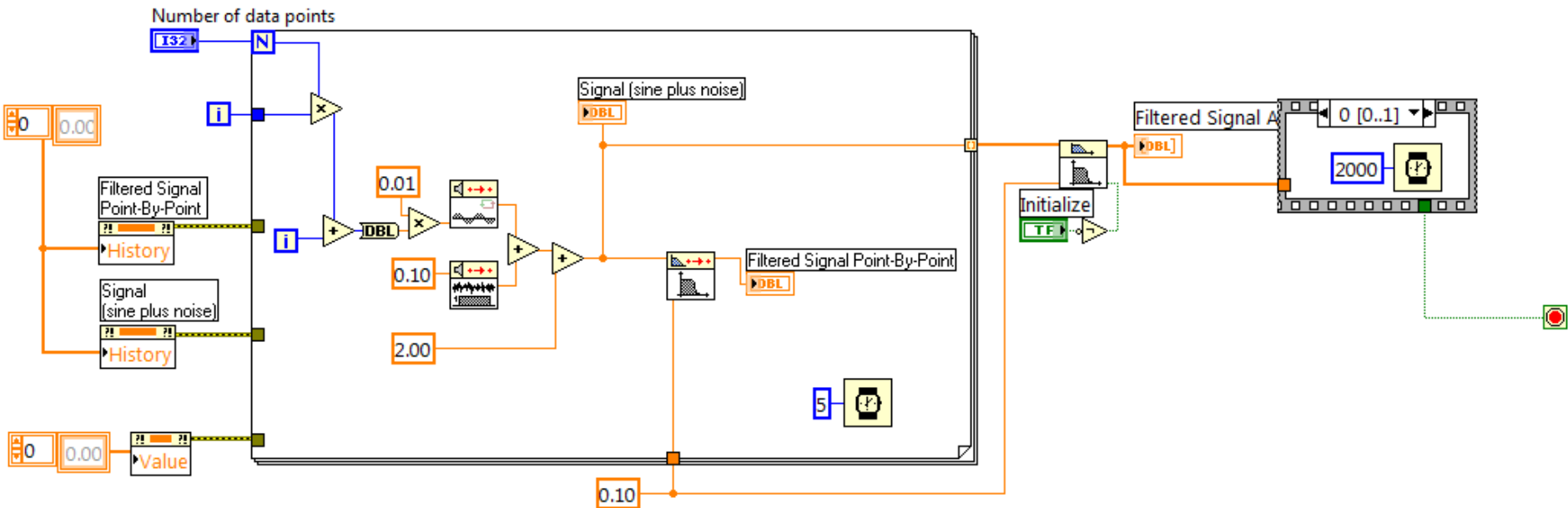
# Adquisición de datos en formato raw



## DAQmx Read (VI)

Lee muestras de la tarea o canales virtuales especificados. Las instancias de este VI polimórfico especifican el formato de las muestras que se devuelven, si la lectura es de una sola muestra o múltiples muestras a la vez y si leen uno o múltiples canales.





## Analog 1D Wfm NChan 1Samp

Lee una forma de onda que contiene una sólo muestra de cada canal de la tarea.



**data** Se devuelve una matriz de formas de onda de una sola dimensión (1D). Cada elemento de la matriz se corresponde con un canal en la tarea. El orden de los canales en la matriz corresponde al orden en que se agregan los canales a la tarea o al orden de los canales que se especifiquen en la propiedad “Channels to Read”.

NI-DAQmx escala los datos a las unidades de la medida, incluyendo cualquier escala personalizada configurada en los canales. Se debe usar el DAQmx Create Virtual Channel VI o el DAQ Assistant para especificar estas unidades.

## Analog 1D DBL NChan 1Samp

Lee una sola muestra en formato punto flotante de cada canal de la tarea que puede contener uno o más canales de entrada analógicos.



**data** Devuelve una **matriz 1D** de muestras. Cada elemento de la matriz corresponde a un canal en la tarea. El orden de los canales en la matriz corresponde al orden en el que se agregan los canales a la tarea o al orden de los canales que se especifique con la propiedad del Canal (Channel to Read property)

NI-DAQmx **escala** los datos a las unidades de medida, incluida cualquier escala personalizada que se configure en los canales. Se usa el DAQmx Create Virtual Channel VI o el DAQ Assistant para especificar estos valores.

## Raw 1D U16

Lee una o más muestras enteras sin signo de 16 bits de una tarea. Se recomienda usar esta instancia de la función DAQmx.Read con dispositivos que usen enteros sin signo de 16 bits como representación interna de sus muestras.

[U16]

**data** devuelve una matriz 1D de muestras enteras sin signo de 16 bits sin procesar.

## Raw 1D I16

Lee una o más muestras enteras con signo de 16 bits de una tarea. Se recomienda usar esta instancia de la función DAQmx.Read con dispositivos que usen enteros con signo de 16 bits como representación interna de sus muestras.

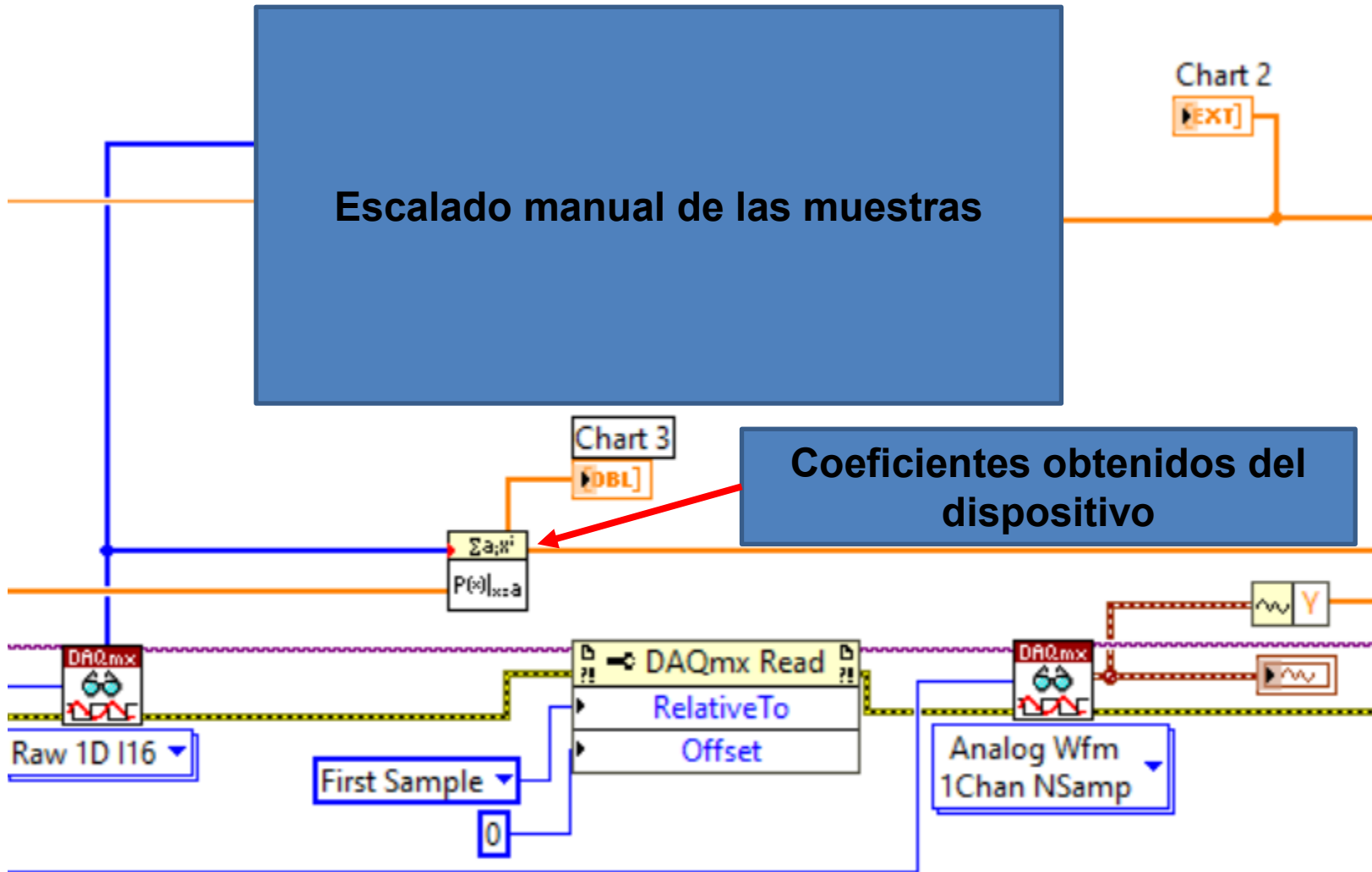
Dada la resolución de su placa y el rango de medición, la siguiente fórmula muestra cómo convertir manualmente los datos binarios adquiridos en datos sin escala (unidades de voltios):

$$\text{Voltage Reading} = (\text{binary reading} / 2^{\text{bit}}) * (V_{\text{max}} - V_{\text{min}})$$

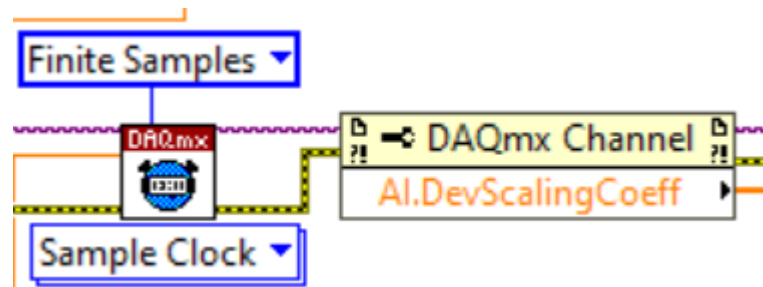
Por ejemplo en una tarjeta de 16 bits de resolución si su rango de voltaje es bipolar (+/- 5V), entonces la lectura es la representación con signo (I16) dentro del rango de -32768 a 32767 para la lectura binaria devuelta. (es decir,  $V_{\text{max}} = 5\text{V}$ ,  $V_{\text{min}} = -5\text{V}$ , bit = 16, lectura binaria = -8192, lectura de voltaje = -1.25V)

Si su rango de voltaje es unipolar (0-10 V), entonces la lectura es la representación sin signo (U16) dentro del rango de 0 a 65535 para la lectura binaria devuelta. (es decir,  $V_{\text{max}} = 10\text{V}$ ,  $V_{\text{min}} = 0\text{V}$ , bit = 16, lectura binaria = 8192, lectura de voltaje = 1.25V)

# Conversión datos binarios a datos escalados



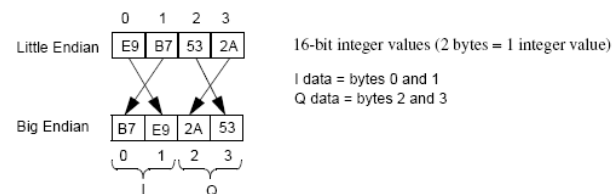
Obtención de coeficientes de escalado del dispositivo a través del nodo de propiedades DAQmx Channel.



El modo raw usa el formato nativo del dispositivo. Los datos se leen directamente desde el dispositivo o el búfer sin escalar ni reordenar. El formato nativo de un dispositivo puede ser un entero de 8, 16 o 32 bits, con signo o sin signo.

Si usa un tamaño entero diferente al formato nativo del dispositivo, un entero puede contener múltiples muestras o una muestra puede extenderse a través de múltiples enteros. Por ejemplo, si usa enteros de 32 bits, pero el dispositivo usa muestras de 8 bits, un entero contiene hasta cuatro muestras. Si usa enteros de 8 bits, pero el dispositivo usa muestras de 16 bits, una muestra puede requerir dos enteros. Este comportamiento varía de un dispositivo a otro. Se debe consultar la documentación de su dispositivo para obtener más información.

**NI-DAQmx no separa los datos sin procesar en canales. Devuelve datos en una matriz 1D intercalada o no intercalada, según el orden sin procesar del dispositivo. Consulte la documentación de su dispositivo para obtener más información.**



Debido a que normalmente hay 2 bytes por muestra, el entero con signo de 16 bits es el tipo de datos numérico más simple y eficiente en la memoria para representar estos datos.

En adquisiciones continuas, la velocidad a la que LabVIEW puede recuperar datos del búfer de adquisición y luego transmitirlos al disco es crucial. Debe poder leer y transmitir los datos lo suficientemente rápido para que el dispositivo DAQ no intente sobrescribir los datos no leídos en el búfer circular.

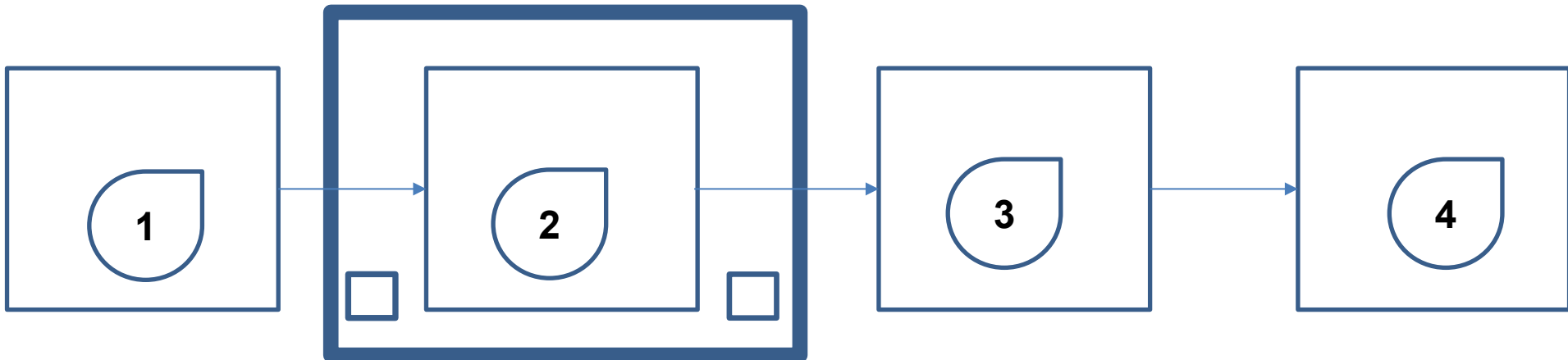
Para **aumentar la eficiencia** de la recuperación de datos, debe evitar ejecutar otras funciones, como las funciones de análisis, mientras la adquisición está en progreso. Además, se puede configurar DAQmx Read para devolver datos binarios sin formato en lugar de datos de voltaje (seleccionando uno de los modos de datos **Raw**). Esto aumenta la eficiencia de la recuperación, así como la transmisión.

Cuando configura DAQmx Read para producir solo datos binarios, puede devolver los datos más rápidamente al búfer que si se utilizara los modos de forma de onda analógica o voltaje. Una **desventaja** de leer y transmitir datos binarios es que los usuarios de otras aplicaciones no pueden leer fácilmente el archivo.



Una operación típica de E/S en ficheros estaría compuesta de los siguientes pasos:

1. Crear o abrir un fichero. Tras la apertura del fichero se obtiene un identificador único del fichero denominado refnum.
2. Se realizar la escritura o lectura del fichero.
3. Cierre del fichero.
4. Comprobación de errores



En LabVIEW se pueden crear ficheros con los siguientes formatos:

**Binario.** Fichero base del resto de formatos de ficheros.

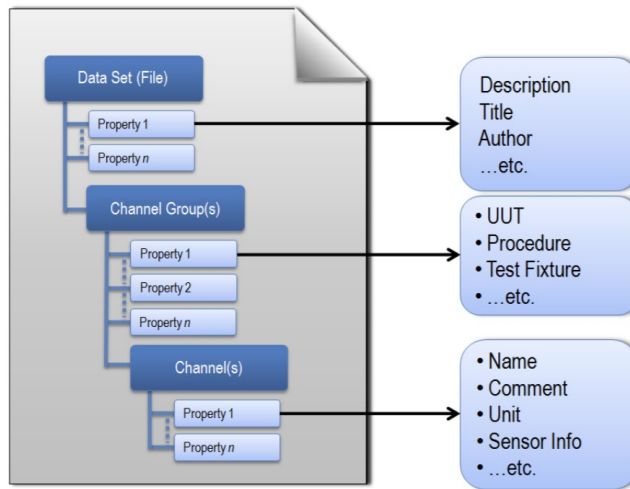
**ASCII.** Un fichero ASCII es un tipo específico de fichero binario en el que se utilizan la codificación ASCII. También se denominan fichero de texto.

**LVM.** Fichero de datos de medición de LabVIEW delimitado con tabulaciones. Se puede abrir con editores de texto. El fichero .lvm incluye información sobre los datos, como la fecha y la hora. Este formato de fichero ASCII creado por National Instrument para LabVIEW.

**TDMS.** Este formato de fichero es un tipo específico de National Instrument.

El formato de archivo binario **TDMS** es un formato de archivo fácilmente intercambiable, estructurado y con capacidad de transmisión de alta velocidad que, cuando se combina con otras tecnologías de NI puede buscar rápidamente sin la necesidad de un diseño, arquitectura o mantenimiento de base de datos complicado y costoso.

Los archivos **TDMS** organizan los datos en una jerarquía de objetos de tres niveles. El nivel superior se compone de un único objeto que contiene información específica del archivo, como, por ejemplo, el autor o título. Cada archivo puede contener un número ilimitado de grupos y cada grupo puede contener un número ilimitado de canales.



```

My Generic Measurement File
Test_Operator  derrick
write_date     2009/04/05
write_time    18:01:54.604
description    This file contains voltage data from five thermocouples.

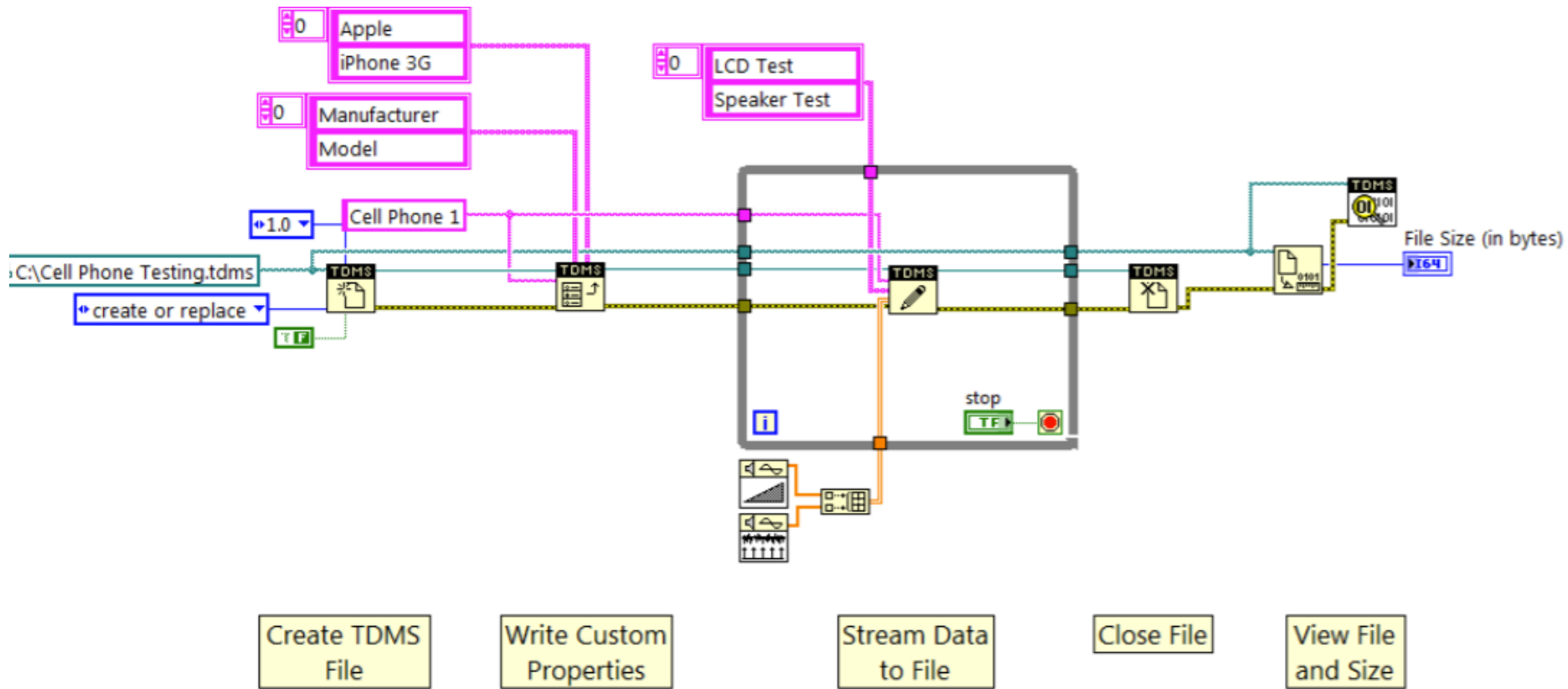
***End_of_Header***

Channels
Samples 100 100 100 100 100
Thermocouple_Type J J J J J
Channel_Max 8.620258 11.523453 21.440928 9.585559
Channel_Min 0.034181 0.565203 1.598865 2.285531
Date 2009/08/05 2009/08/05 2009/08/05 2009/08/05 2009/08/05
Time 18:01:54 18:01:54 18:01:54 18:01:54 18:01:54
Y_Unit_Label volts volts volts volts volts
X_Dimension Time Time Time Time Time
XD 0.000E+0 0.000E+0 0.000E+0 0.000E+0 0.000E+0
delta_X 0.001 0.001 0.001 0.001 0.001

***End_of_Header***

voltage_0 voltage_1 voltage_2 voltage_3 voltage_4
0.034181 0.565203 1.598865 2.285531 3.078097
0.508438 1.557054 2.273224 2.876675 3.936277
1.274453 1.919004 2.690512 3.775750 4.434034
1.986755 2.779321 3.187963 4.092532 4.682150
2.648091 3.437300 3.848079 4.966277 5.576342
2.864162 3.930174 4.454176 5.257424 5.866878
3.555712 4.565264 4.990081 5.990173 6.342662
3.877682 4.656514 5.795770 6.304819 7.136448
4.543901 5.275430 6.055177 6.619160 7.080599
5.078280 5.651112 6.559343 6.941130 7.811213
5.679189 6.277047 7.207251 7.364422 7.993713
6.457411 6.583148 7.674184 8.105716 8.107242
    
```

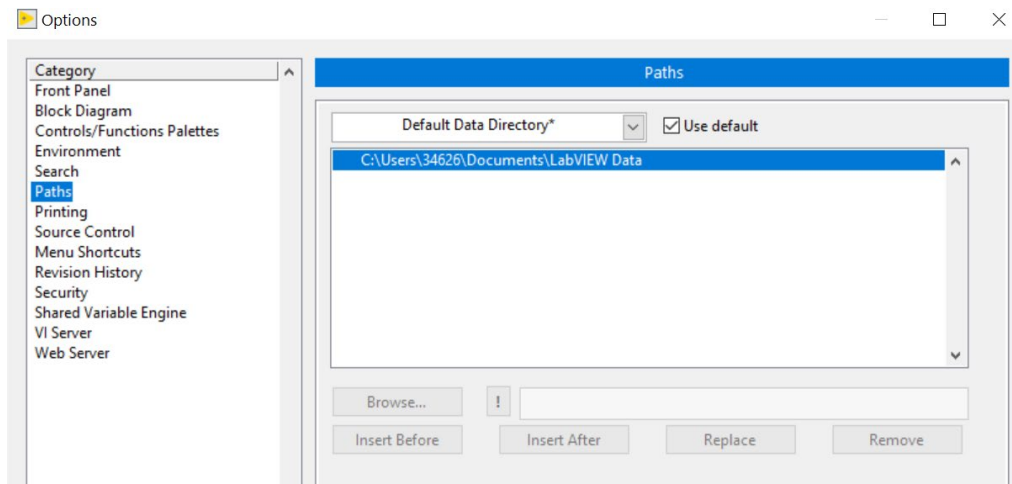
# Edición de fichero TDMS



LabVIEW crea el directorio LabVIEW Data en el directorio de fichero predeterminado del sistema operativo.

De forma predeterminada el VI Express Write LabVIEW measurement File almacena los ficheros .lvm en este directorio, y el VI Express Read LabVIEW Measurement File lee de ese directorio. La constante “Default Data Directory” y la propiedad “Default Data Directory” pueden ser usadas para consultar el directorio Data de LabVIEW.

Se puede modificar el directorio por defecto a través del menú Tools-> Options->Path y seleccionando en el desplegable la opción “Default Data Directory”.



Algunos de los VIs de E/S en fichero permiten realizar los 3 primeros pasos del proceso de E/S: abrir, leer/escribir y cerrar. Si un VI realiza los tres pasos se denomina VI de **alto nivel**. Estos VIs no son eficientes como los VIs de bajo nivel disponibles también en LabVIEW. Se recomienda el uso de los VIs de bajo nivel cuando se accede a un fichero en un bucle.

Los VIs de alto nivel incluidos en LabVIEW son los siguientes:

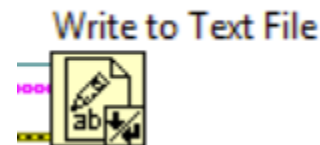
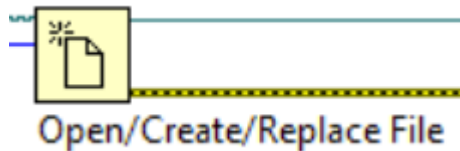
- **Write to Spreadsheet File:** convierte un array 2D y 1D de números de doble precisión en una cadena de caracteres de texto y escribe ésta en un fichero ASCII. El VI permite crear o abrir un fichero, escribir en él y cerrarlo después.
- **Read From Spreadsheet File:** Lee un número concreto de líneas o filas de un fichero de texto numérico empezando por un desplazamiento de caracteres concreto y convierte los datos en un array 2D de número de doble precisión. El VI abre el fichero, lee y lo cierra.
- **Write to Measurement File:** escribe datos en un formato de fichero de medidas basado en texto (.lvm) o en un fichero de medidas binario (.tdms). Se puede especificar el método de guardado, el formato de fichero, el tipo de encabezado y el delimitador.
- **Read from Measurement File:** Permite leer de un fichero .lvm o .tdms.

# Entrada/salida de fichero de bajo nivel

Los VIs de E/S de fichero de bajo nivel realizan sólo una etapa del proceso de E/S. Existe una función para abrir un fichero ASCII, otra para leer o escribir y otra cerrarlo.

El uso de operaciones de E/S de bajo nivel reduce el número de veces que se interactúa con el sistema operativo para abrir y cerrar el fichero.

Para evitar abrir y cerrar el fichero se debe pasar un refnum al fichero en el bucle



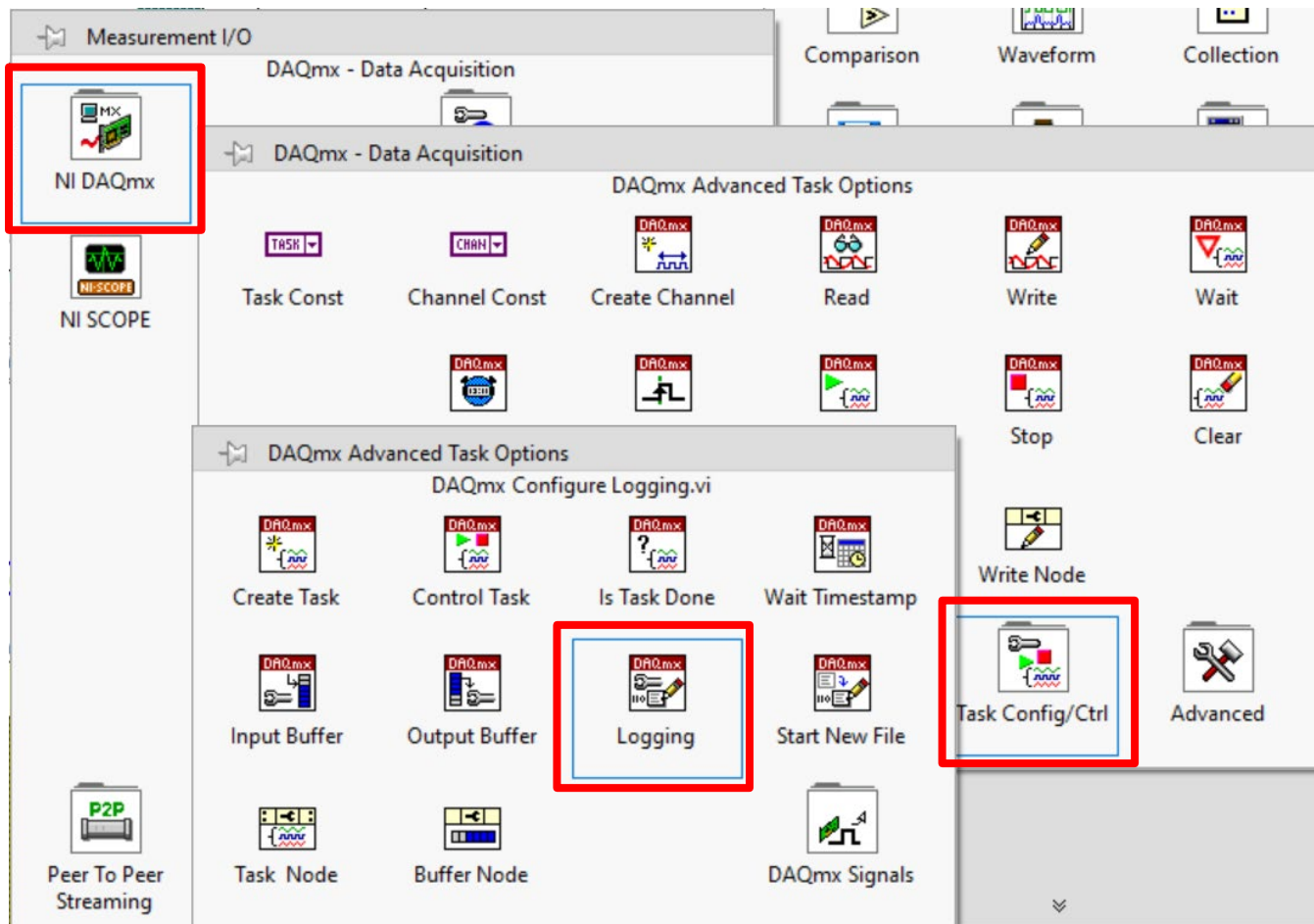
NI-DAQmx 9.0 o posterior proporciona un nuevo VI, **DAQmx Configure Logging VI**, que permite a los usuarios configurar la escritura de datos en un archivo TDMS dentro de un solo VI en la paleta DAQ. Este VI permite transmitir los datos directamente desde el búfer del dispositivo al disco disco duro.

DAQmx Configure Logging (TDMS) .vi no incluye la función de transmitir al disco duro de forma selectiva.

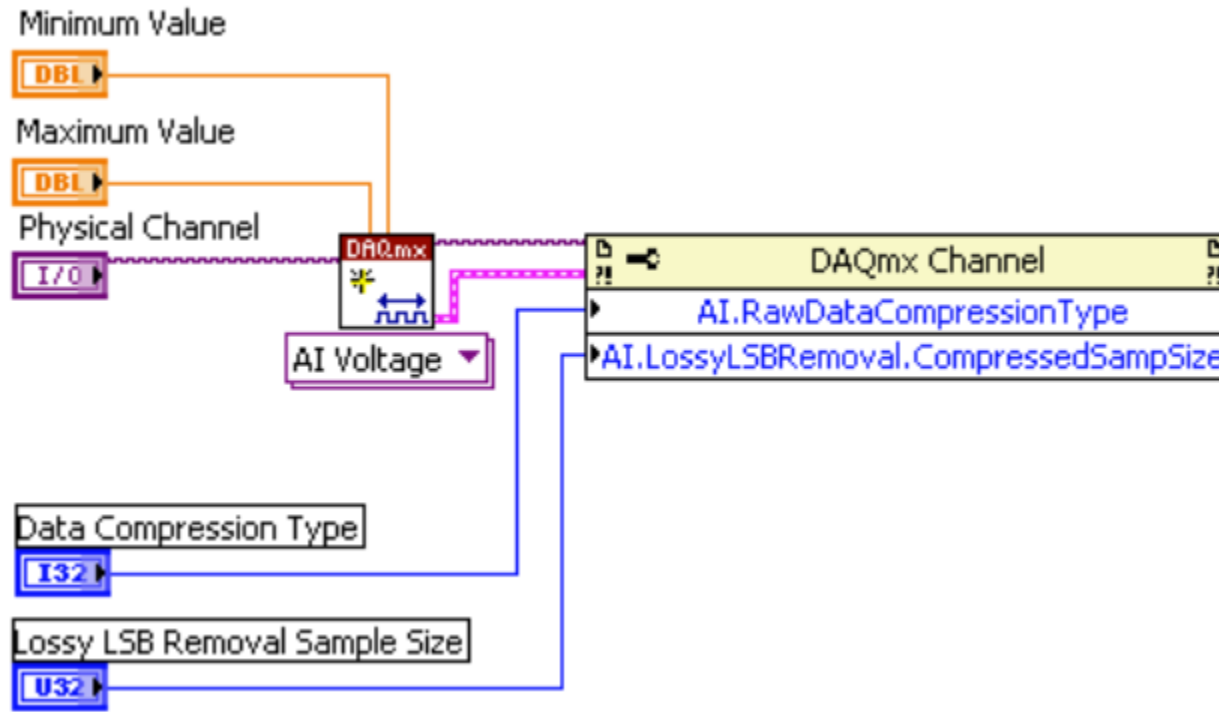
Las marcas de tiempo no se guardan. En los metadatos se indica la hora de inicio (T0) y una frecuencia de muestreo (dt), que permiten calcular el tiempo de cada muestra.



# Escritura de alta velocidad NI-DAQmx a Disco



NI-DAQmx implementa una forma extremadamente simple de **compresión de datos**. Se descartan los bits de mayor peso que no son utilizados. Por ejemplo, un canal con resolución de 12 bits y un tamaño de muestreo de 16 bits contiene cuatro bits no utilizados. Se puede, por tanto, descartar los cuatro bits sin pérdida de información reduciendo el espacio en disco requerido en un 25%.



# GENERACIÓN DE SEÑALES

En LabVIEW se puede generar señales para probar, por ejemplo algoritmos, cuando no tenemos señales reales disponibles. Las señales se pueden generar haciendo uso de fórmulas matemáticas, arrays the puntos o usando los Vis de generación de señales comunes.

Functions → Programming → Waveforms → Analog Waveform → Generation



Waveform Generation

Formula Waveform.vi

Formula Wfm

Basic FuncGen    Tones & Noise

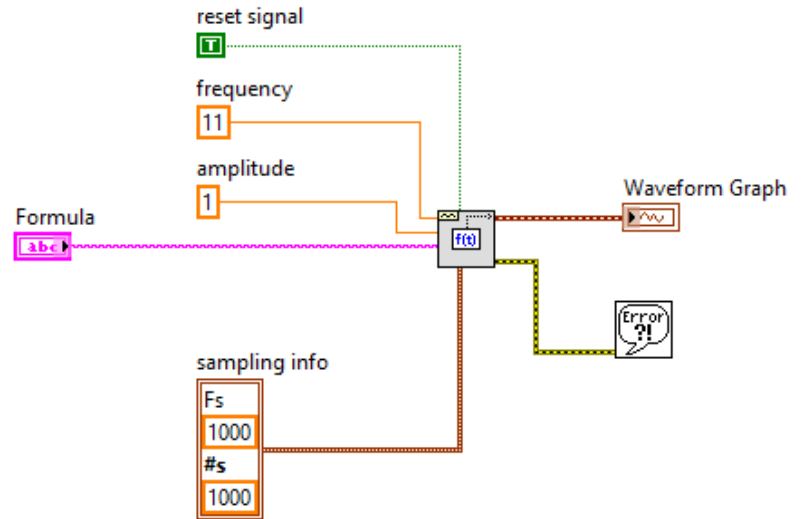
Sine Wfm    Square Wfm    Triangle Wfm    Sawtooth Wfm

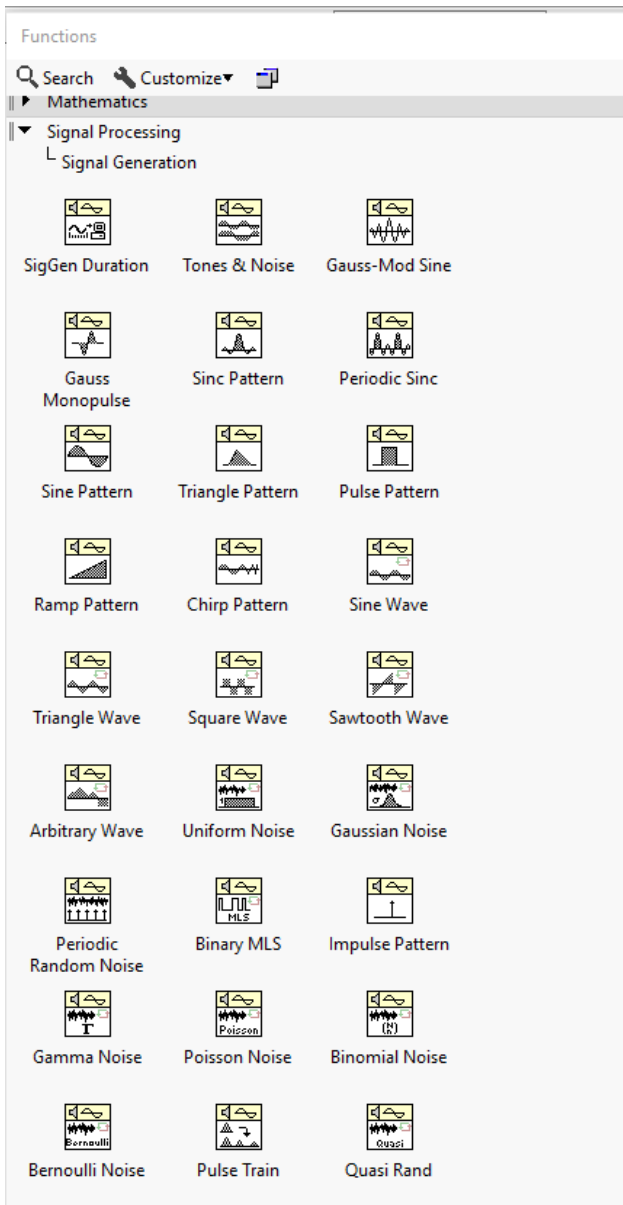
Basic Multitone    Multi with Amps    Multitone Gen

Uniform Wfm    Gaussian Wfm    PRN Wfm    Inv f Wfm

Gamma Wfm    Poisson Wfm    Binomial Wfm    Bernoulli Wfm    MLS Wfm

Simulate Sig    Sim Arb Sig





$$\text{Frecuencia normalizada} = \frac{f. \text{analógica}}{f. \text{muestreo}}$$

Cuando se use los siguientes VIs se debe hacer uso de la frecuencia normalizada:

1. Sine Wave
2. Square Wave
3. Sawtooth Wave
4. Triangle Wave
5. Arbitrary Wave
6. Chirp Pattern

Functions

Search  Customize ▾

- ▶ Mathematics
- ▶ Signal Processing
- ▶ Data Communication
- ▶ Connectivity
- ▶ Control & Simulation
- ▼ Express
  - ↳ Input
    - DAQ Assist
    - Instr Assist
    - Simulate Sig**
    - Sim Arb Sig
    - Acquire Sound
    - Read Meas File
    - Prompt User

**Simulate Signal**

- Sine ▶
- error out ▶
- Amplitude ▶
- error in (no error) ▶
- Frequency ▶
- Offset ▶
- Phase ▶

### Configure Simulate Signal [Simulate Signal]

**Signal**

Signal type: Sine

Frequency (Hz): 10.1      Phase (deg): 0

Amplitude: 1      Offset: 0      Duty cycle (%): 50

Add noise

Noise type: Uniform White Noise

Noise amplitude: 0.6      Seed number: -1      Trials: 1

**Timing**

Samples per second (Hz): 1000

Number of samples: 100  Automatic

Integer number of cycles

Actual number of samples: 100

Actual frequency: 10.1

Simulate acquisition timing  
 Run as fast as possible

**Result Preview**

Amplitude vs Time

**Time Stamps**

Relative to start of measurement

Absolute (date and time)

**Reset Signal**

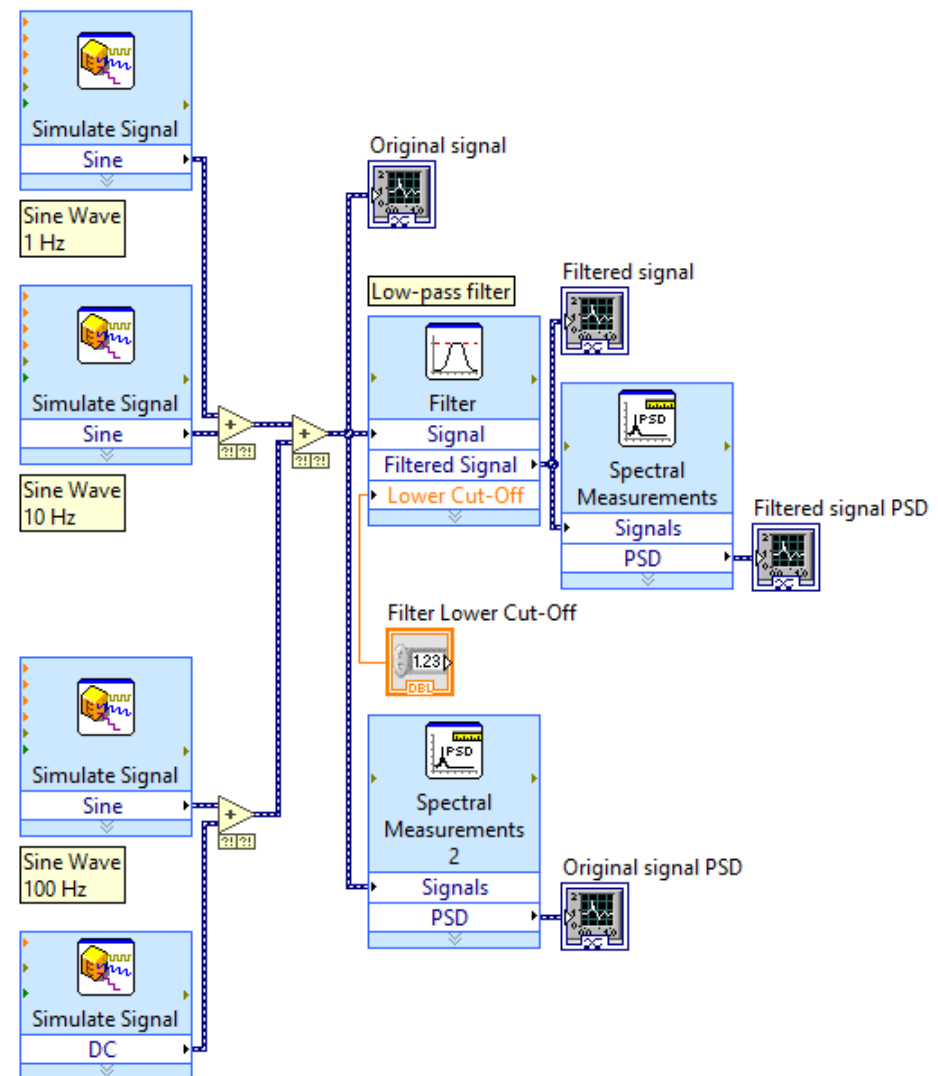
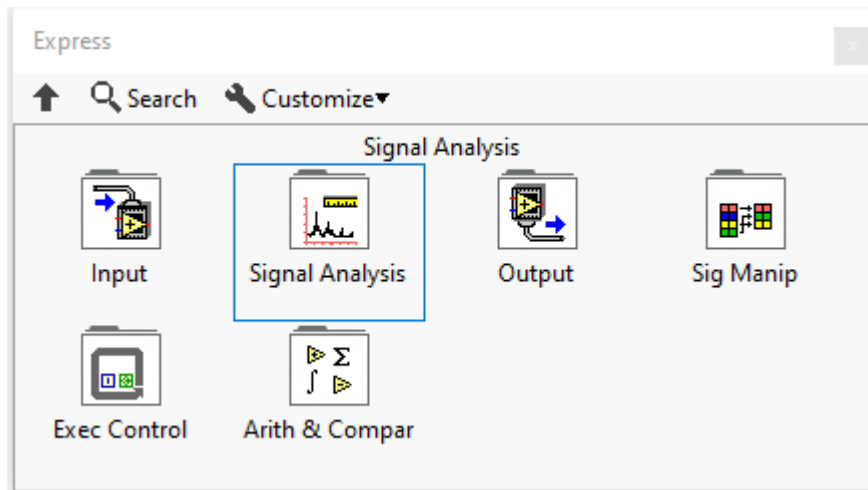
Reset phase, seed, and time stamps

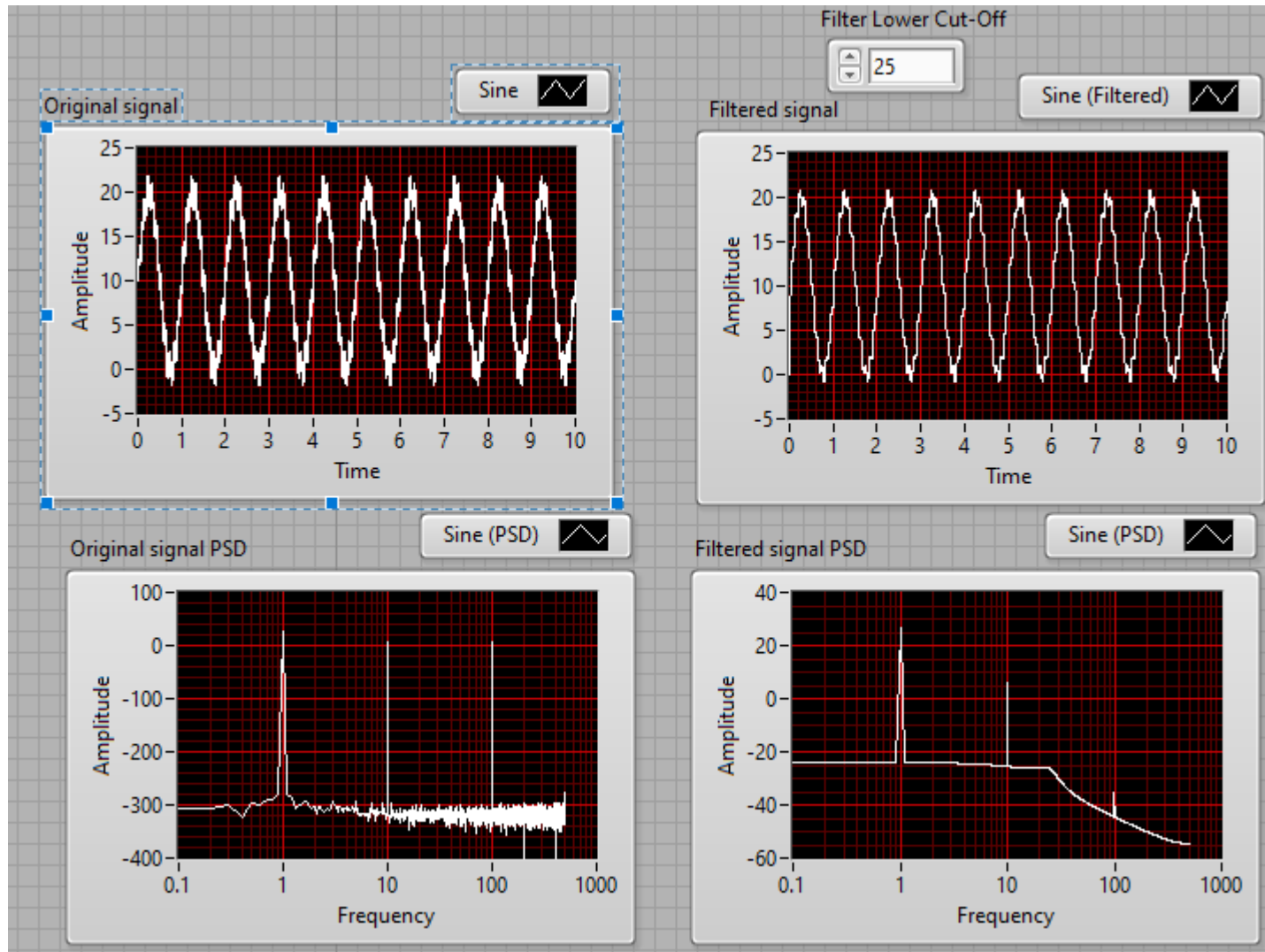
Use continuous generation

**Signal Name**

Use signal type name

Signal name: Sine

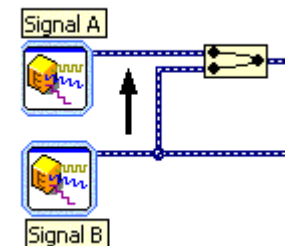
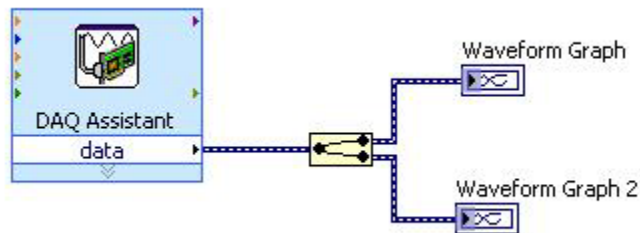




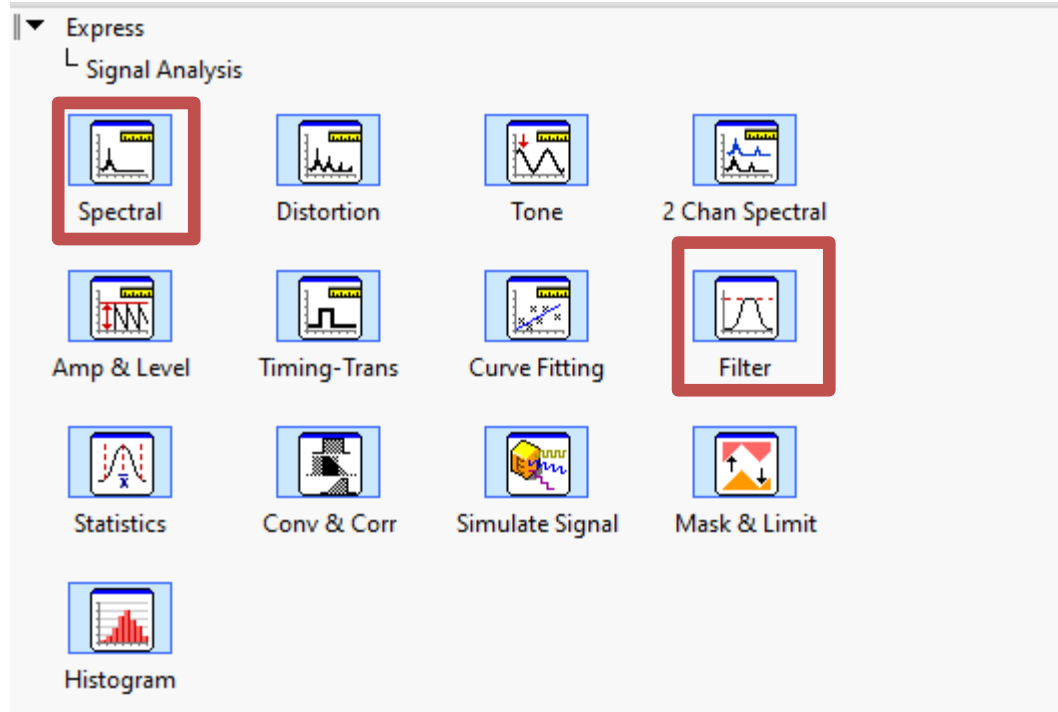
La mayoría de los Express VIs aceptan y/o entregan tipos de datos dinámicos, los cuales se representan con una línea de color azul oscuro.

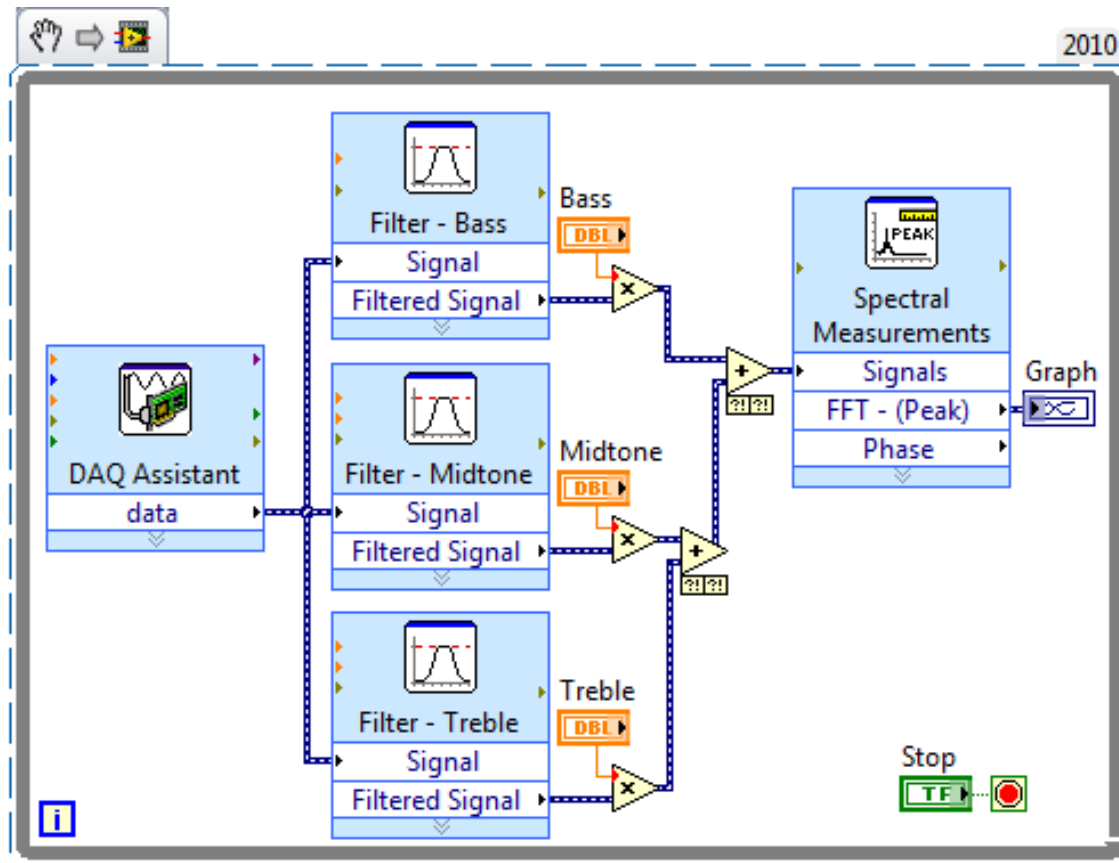
El tipo de datos dinámico es para uso con Express VIs. La mayoría de los VIs y funciones que se incluyen en LabVIEW no aceptan este tipo de datos. Para usar una función o VI integrada para analizar o procesar los datos que incluye el tipo de datos dinámico, se debe convertir el tipo de dato dinámico usando los **Convert to Dynamic Data** y **Convert from Dynamic Data**.

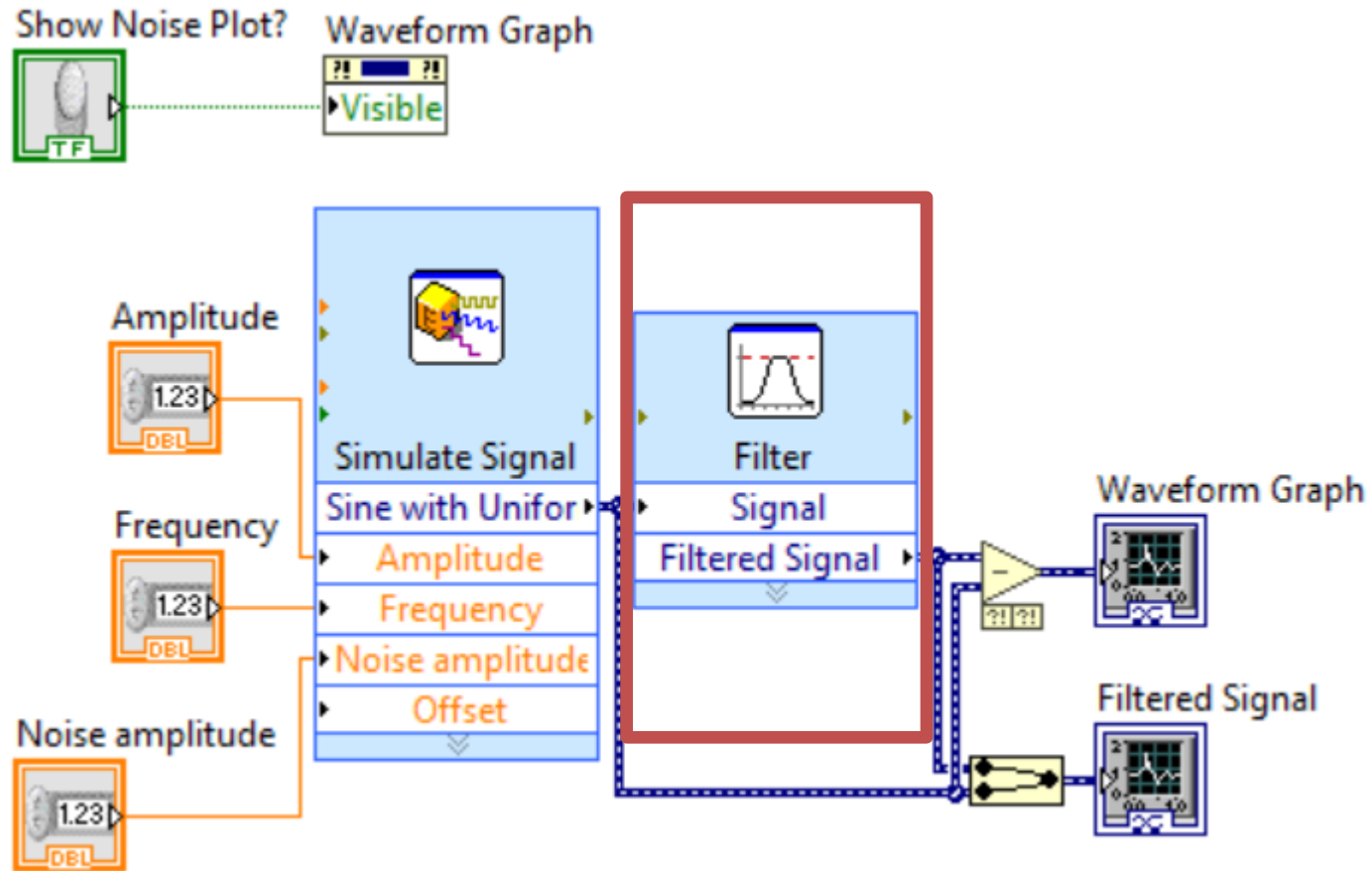
## Dividir y combinar señales de tipo de datos dinámicos

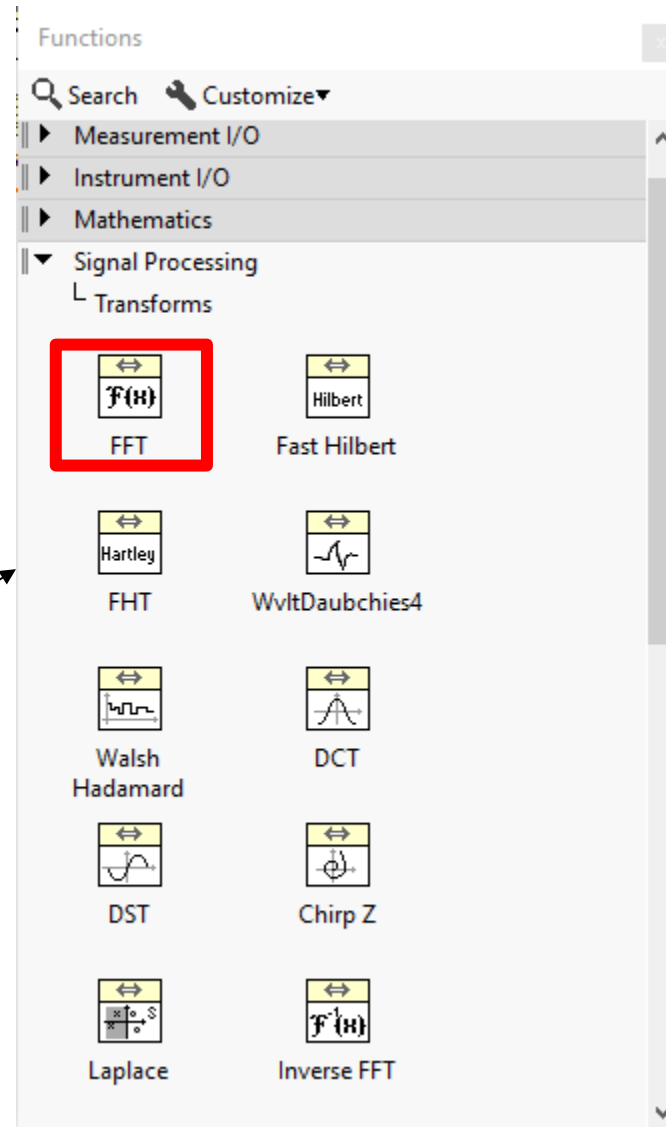
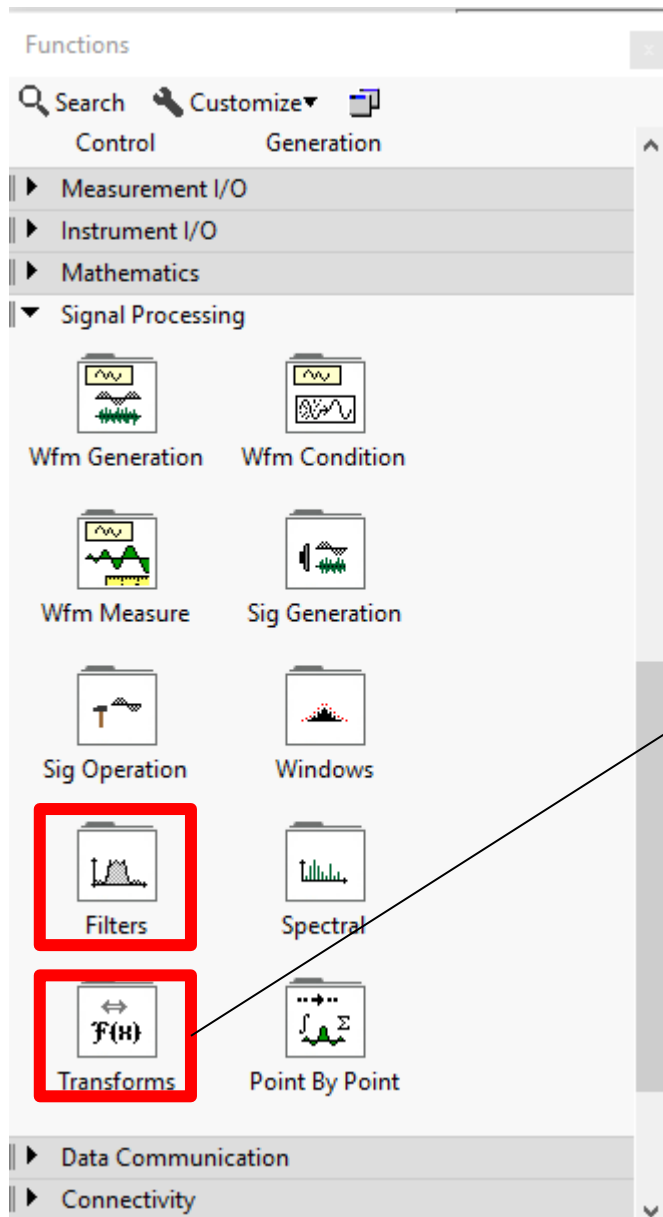


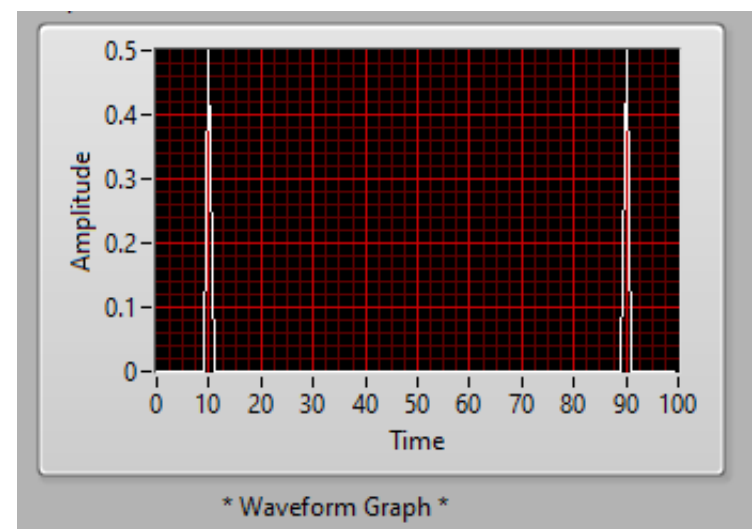
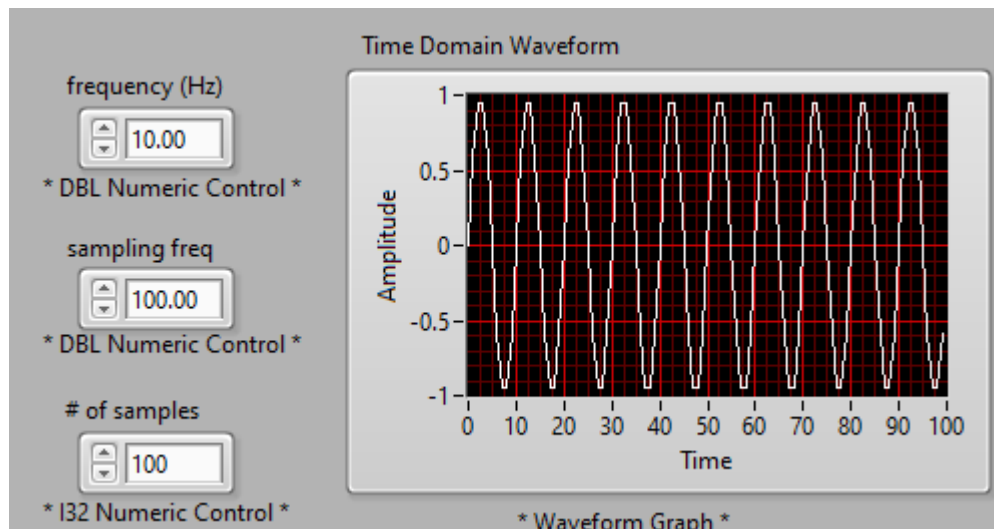
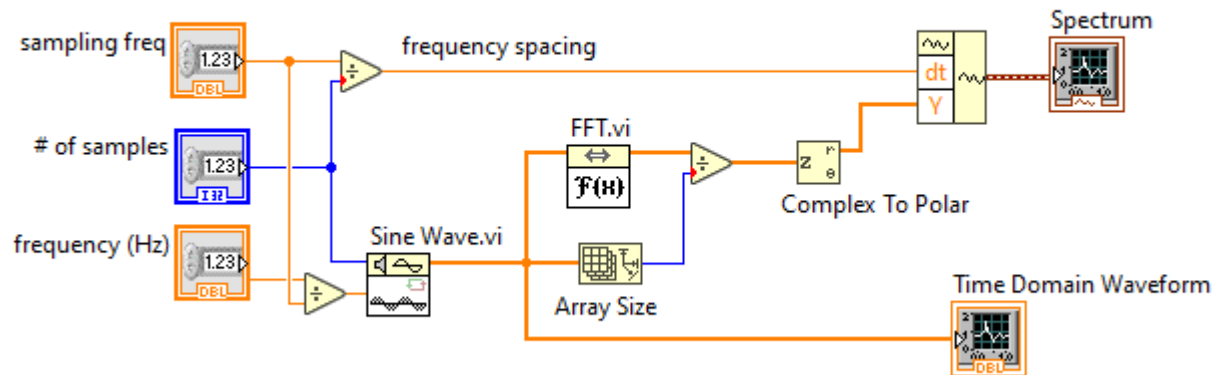
# PROCESADO DE SEÑALES

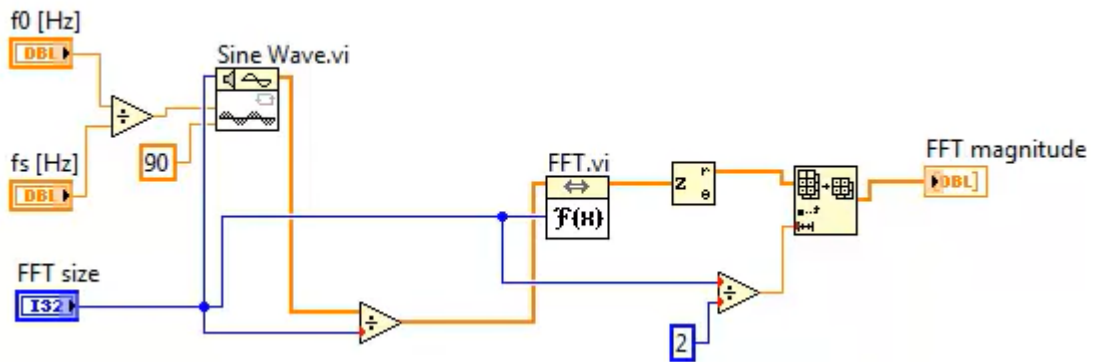
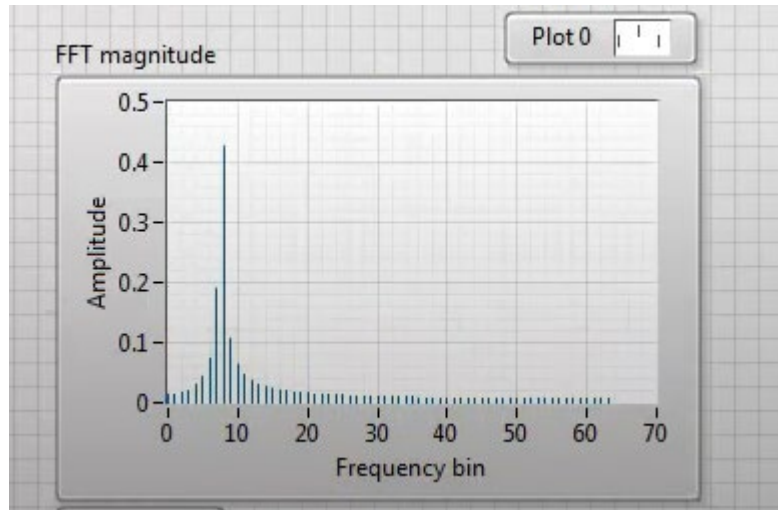




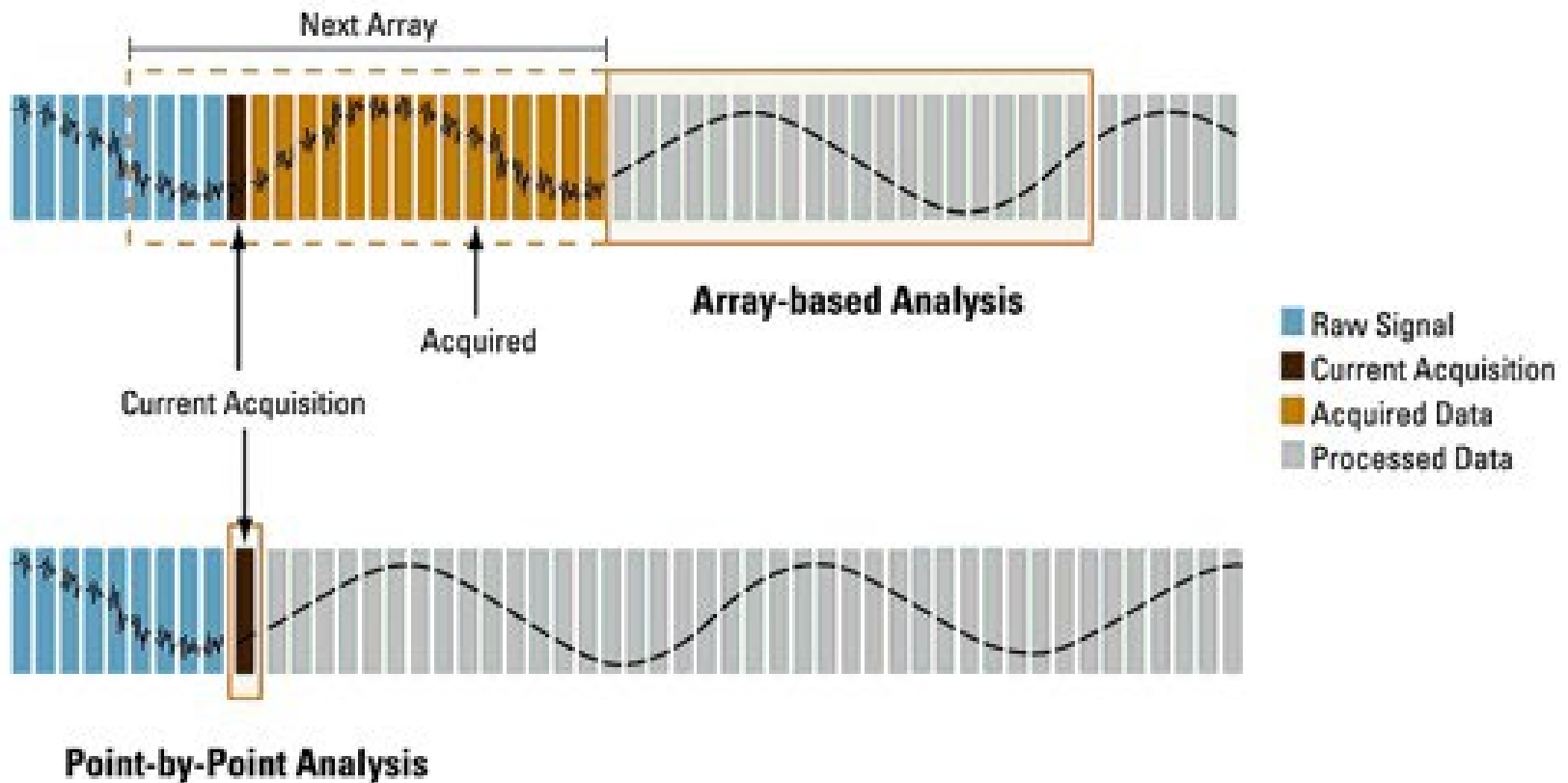




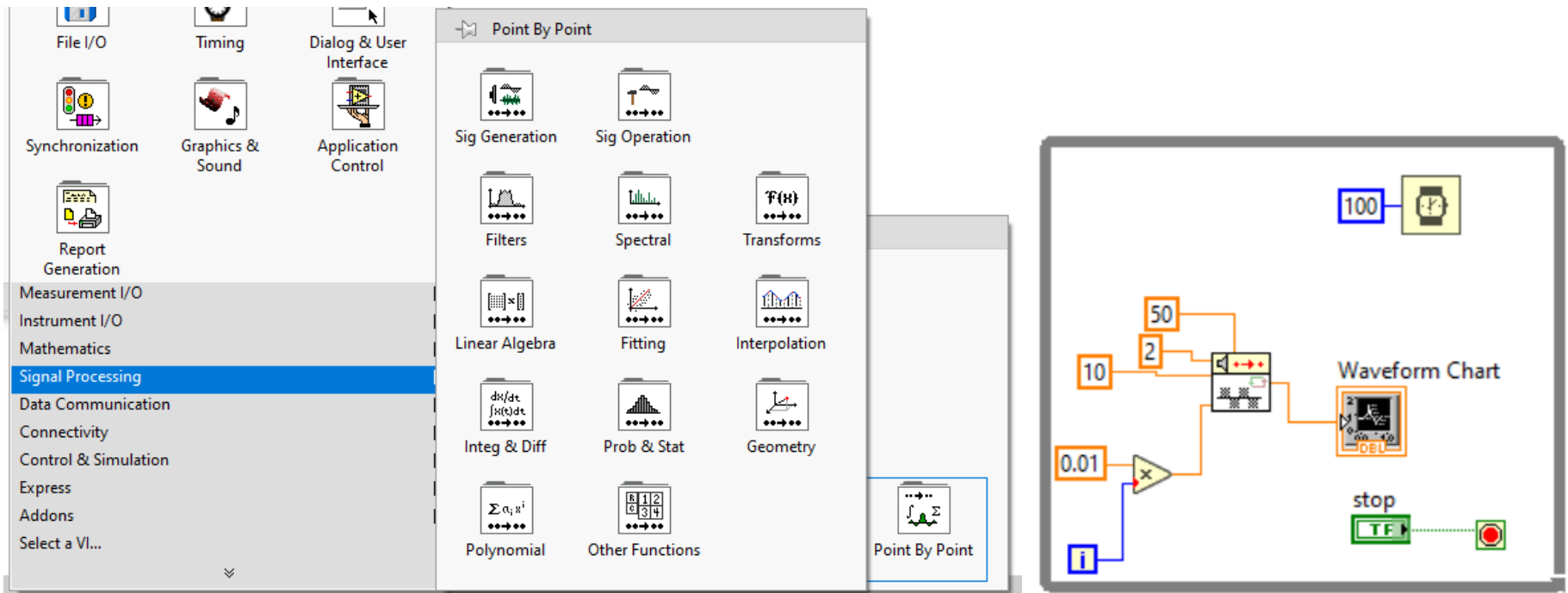




El procesamiento punto por punto es un tipo escalar de procesamiento de datos. El procesamiento punto por punto es adecuado para tareas de procesamiento de datos en tiempo real, como el filtrado de señales, ya que permite la sincronización de las entradas y las salidas. Por otro lado, en el procesamiento basado en arrays, existe un retraso entre la adquisición y el procesamiento de datos.



LabVIEW ofrece rutinas de análisis para ejecución punto por punto; estas rutinas están diseñadas específicamente para cumplir con las necesidades de análisis en línea en aplicaciones en tiempo real



The image shows the LabVIEW software interface. On the left, the 'Signal Processing' palette is open, displaying various functional blocks categorized into:

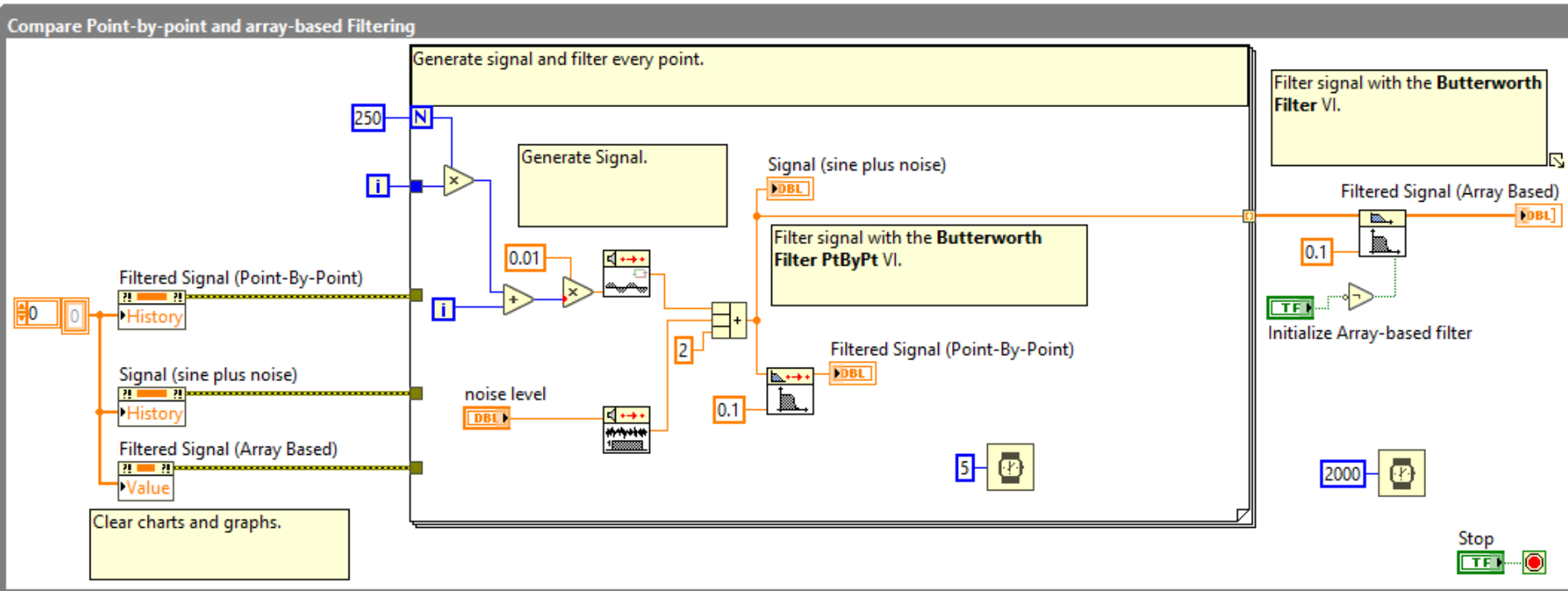
- File I/O
- Timing
- Dialog & User Interface
- Synchronization
- Graphics & Sound
- Application Control
- Report Generation
- Measurement I/O
- Instrument I/O
- Mathematics
- Signal Processing** (highlighted)
- Data Communication
- Connectivity
- Control & Simulation
- Express
- Addons
- Select a VI...

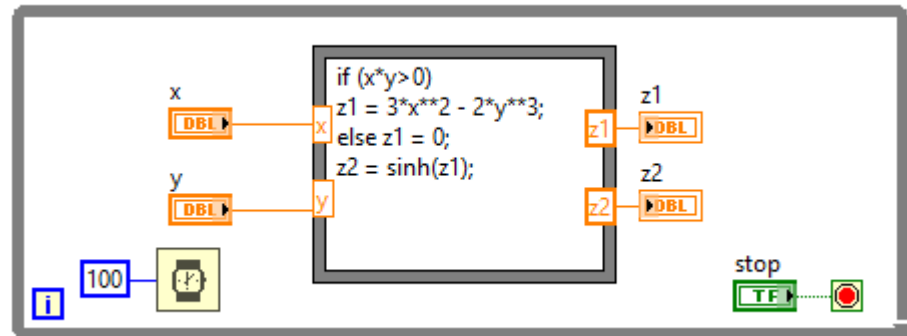
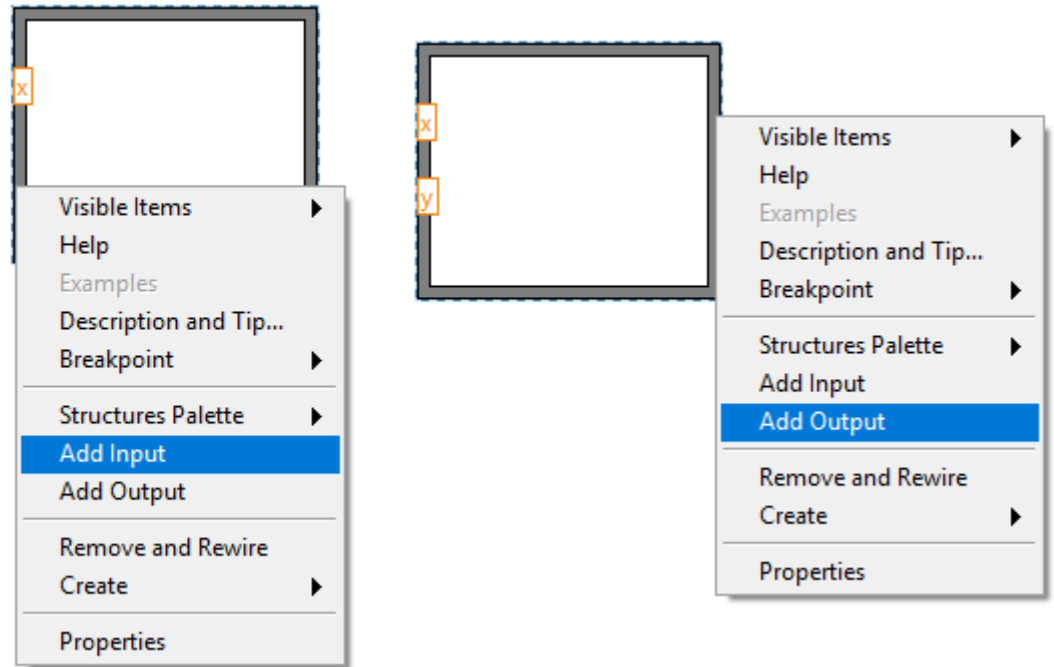
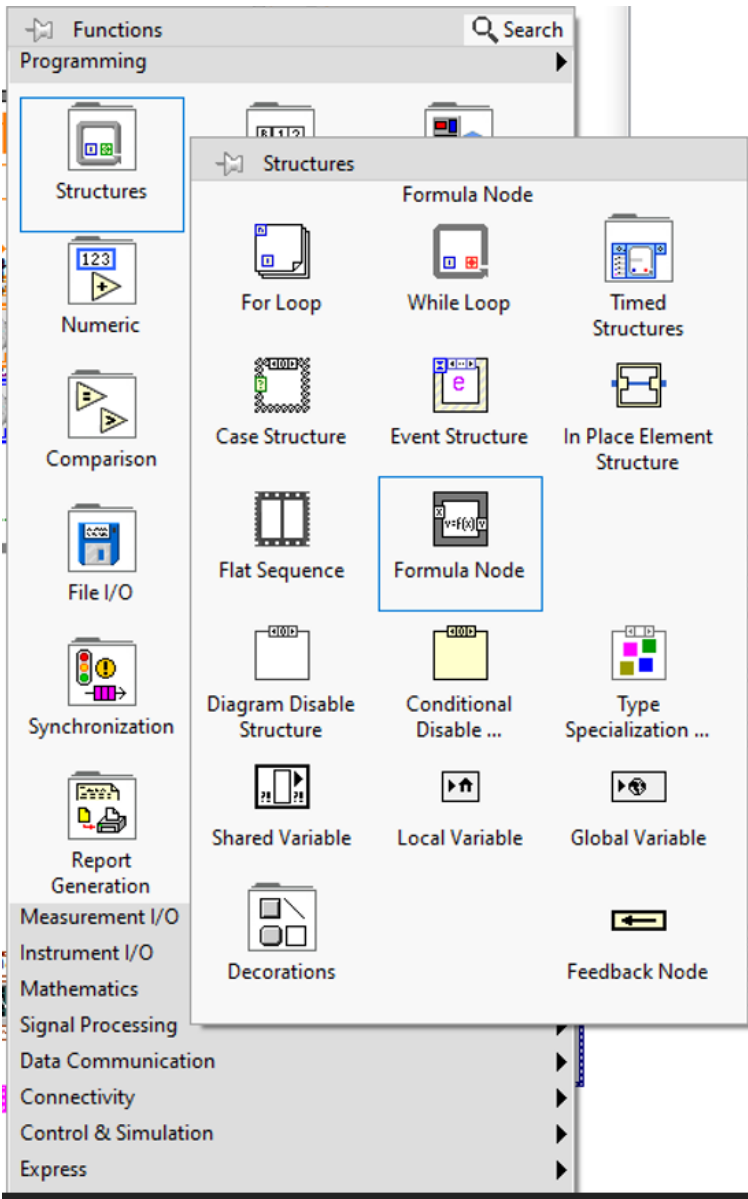
The 'Point By Point' sub-palette is also visible, containing:

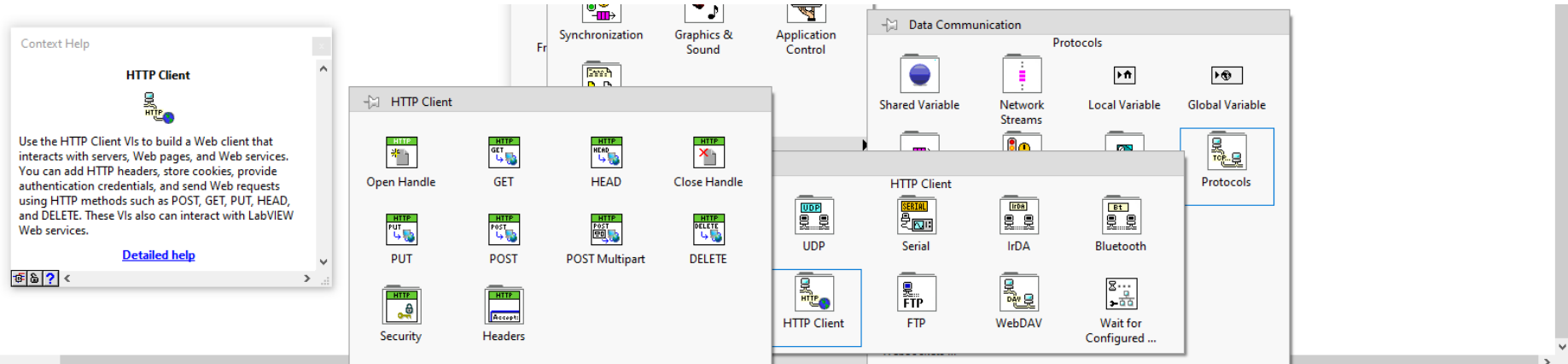
- Sig Generation
- Sig Operation
- Filters
- Spectral
- Transforms
- Linear Algebra
- Fitting
- Interpolation
- Integ & Diff
- Prob & Stat
- Geometry
- Polynomial
- Other Functions

On the right, a block diagram illustrates a signal processing routine. It features a 'Waveform Chart' block with a 'stop' button. The diagram includes numerical constants (10, 50, 2, 0.01, 100) and mathematical operations (multiplication, addition, and a function block labeled 'F(x)'). A 'stop' button is connected to a 'TE' block, which is further connected to a red stop button.

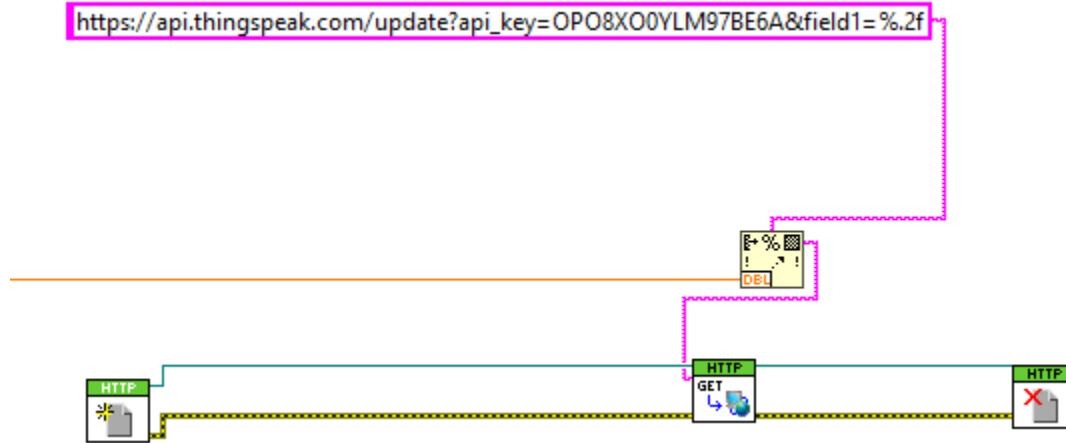
# Procesamiento punto a punto vs array

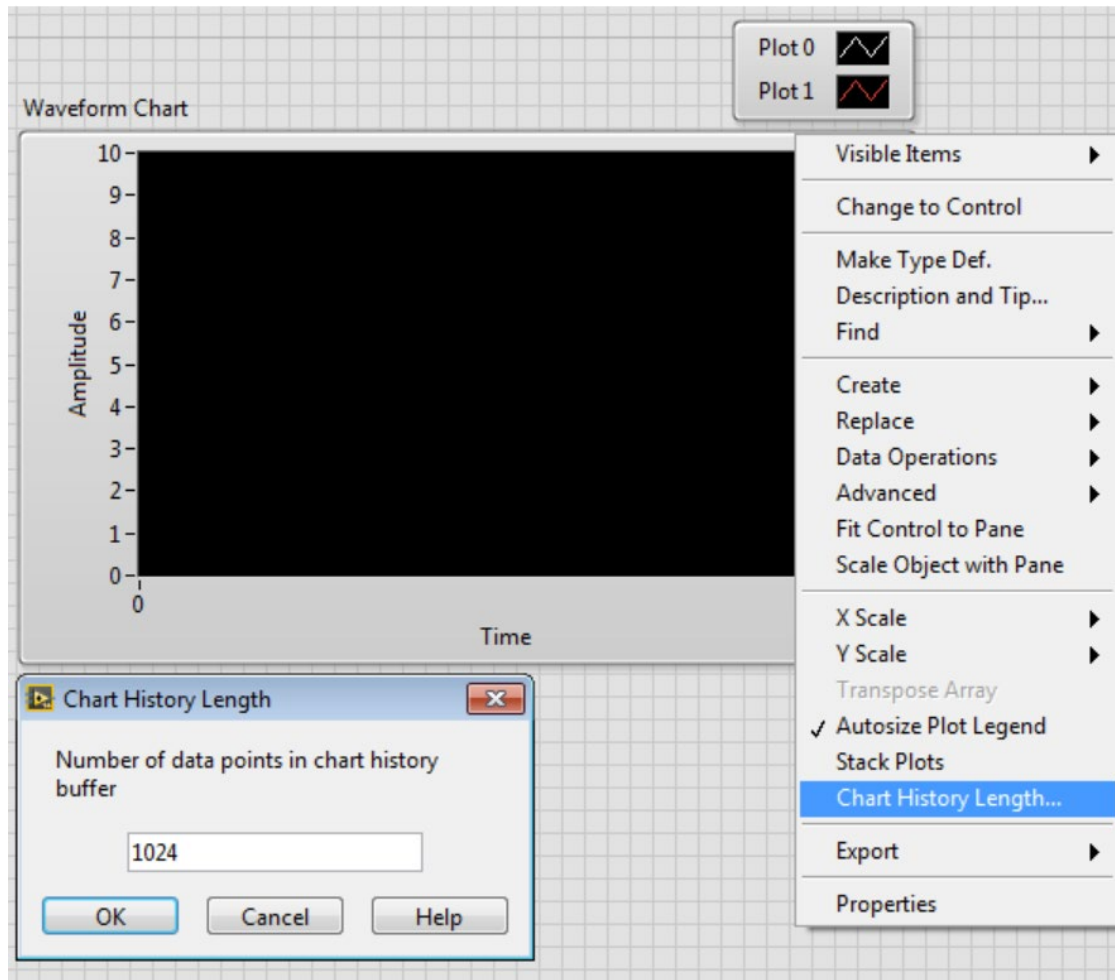






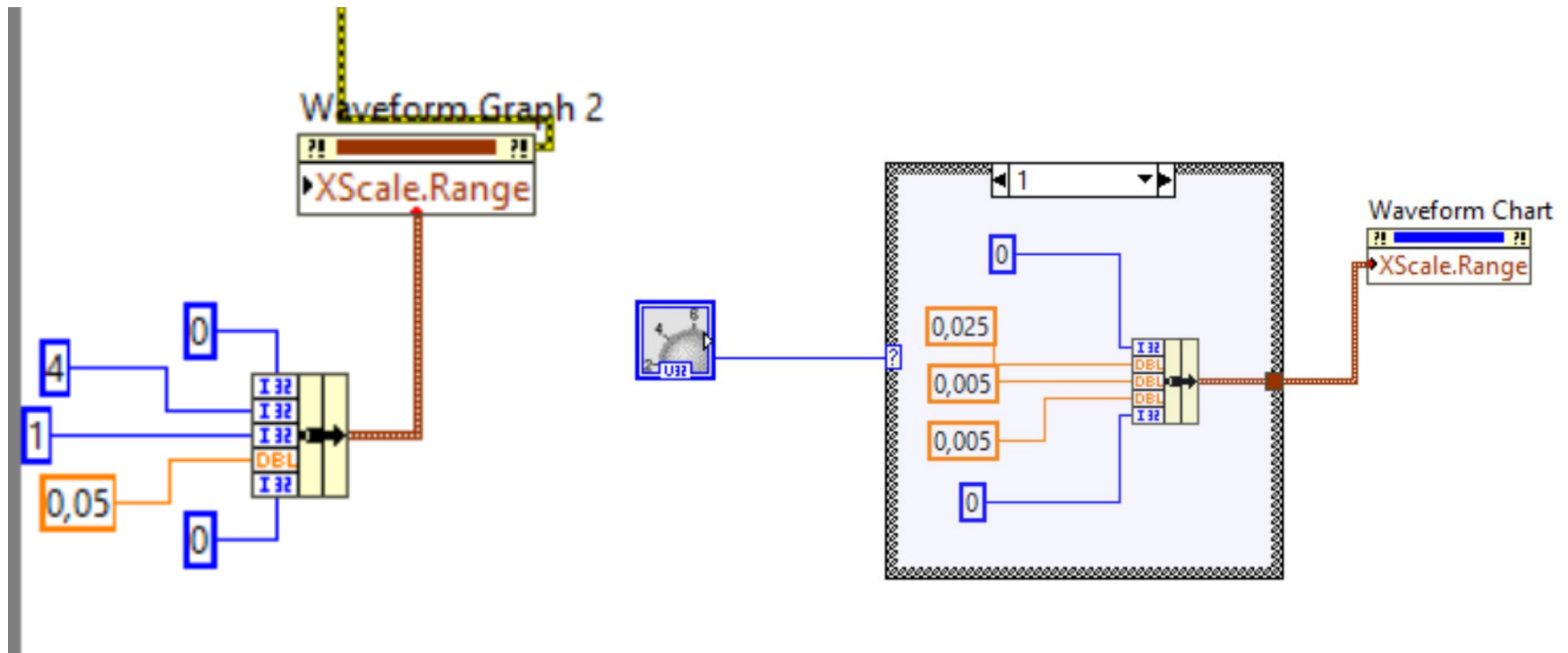
`https://api.thingspeak.com/update?api_key=OPO8X00YLM97BE6A&field1=%2f`



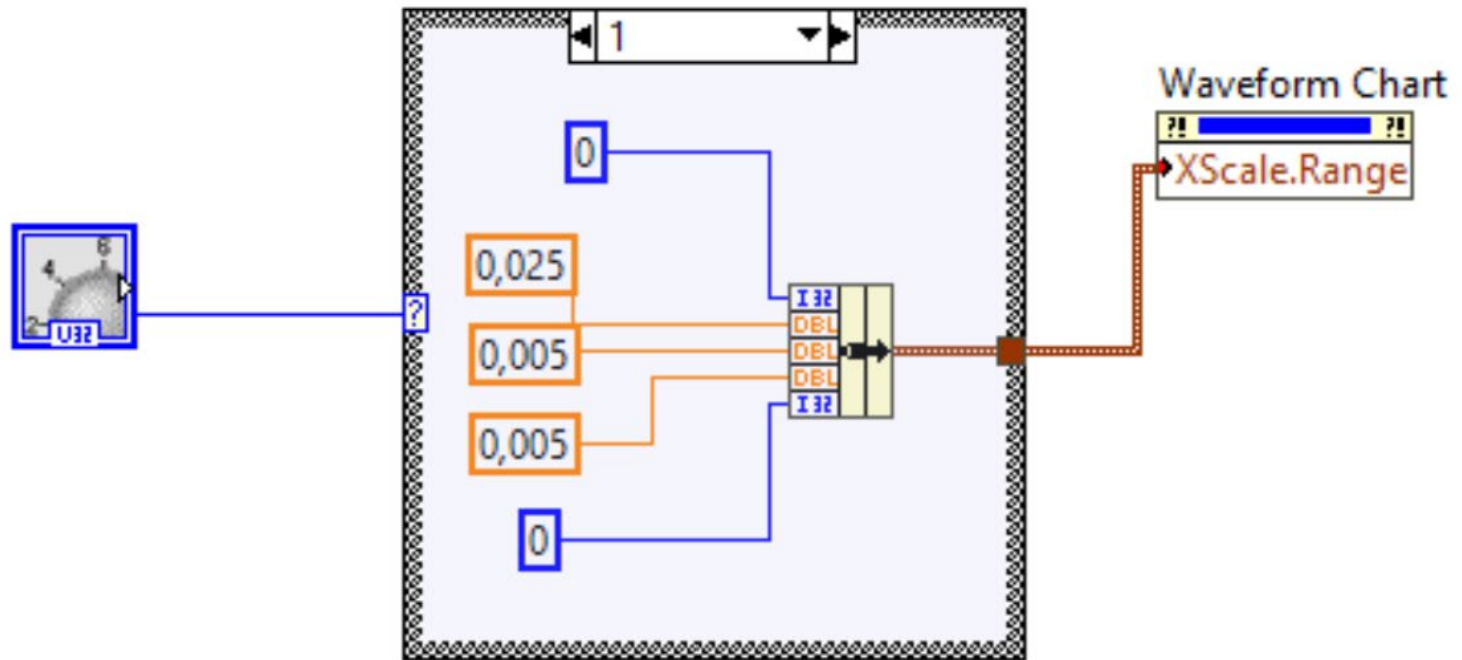


The image shows a software interface for a Waveform Chart. The chart area is currently black, with a vertical axis labeled 'Amplitude' ranging from 0 to 10 and a horizontal axis labeled 'Time'. Above the chart, there are two plot legends: 'Plot 0' with a black waveform icon and 'Plot 1' with a red waveform icon. A context menu is open over the chart area, listing various actions such as 'Visible Items', 'Change to Control', 'Make Type Def.', 'Description and Tip...', 'Find', 'Create', 'Replace', 'Data Operations', 'Advanced', 'Fit Control to Pane', 'Scale Object with Pane', 'X Scale', 'Y Scale', 'Transpose Array', 'Autosize Plot Legend' (checked), 'Stack Plots', 'Chart History Length...' (highlighted), 'Export', and 'Properties'. In the foreground, a dialog box titled 'Chart History Length' is open, containing the text 'Number of data points in chart history buffer' and a text input field with the value '1024'. The dialog box has 'OK', 'Cancel', and 'Help' buttons.

## Property node XScale.Range



# Gráficos. Ajuste de las escalas de los ejes



# Gráficos. Ajuste de las escalas de los ejes

