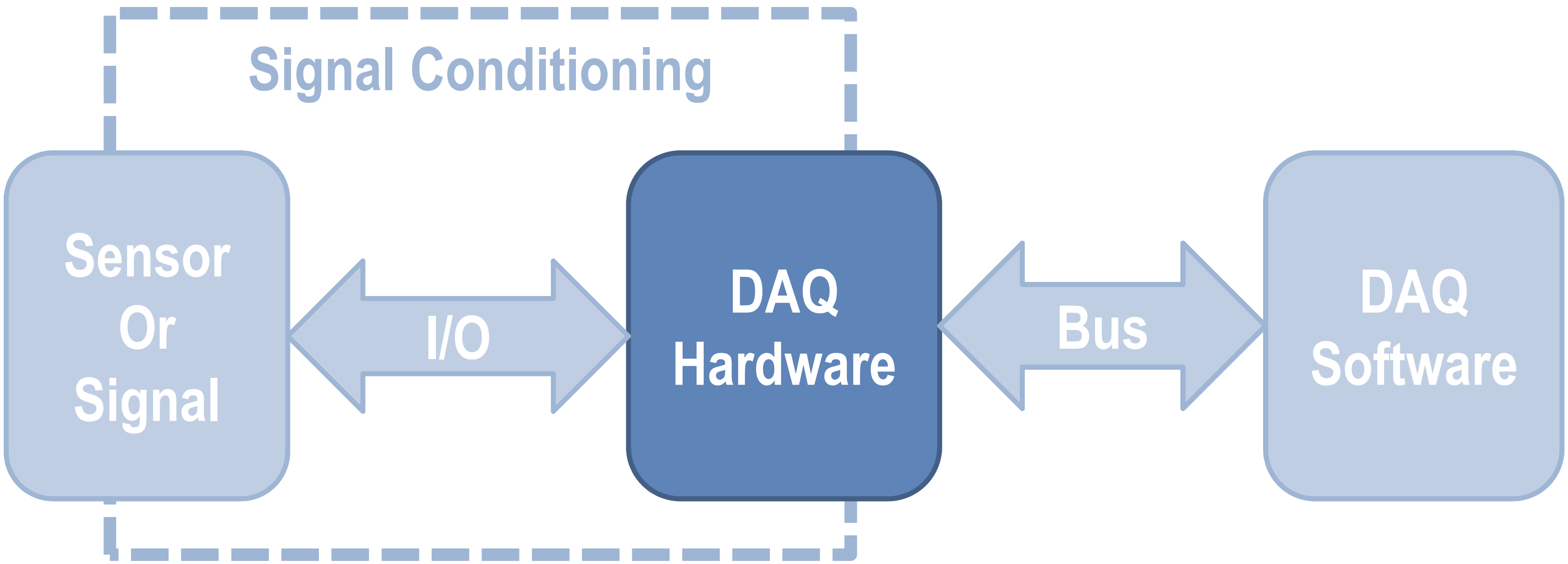
Lesson 2
Data Acquisition Hardware and Software

TOPICS

A. DAQ Hardware Overview
B. Components of a DAQ Device
C. Choosing Appropriate DAQ Hardware
D. DAQ Software Overview
E. Overview of NI-DAQmx VIs

A. DAQ Hardware Overview

- Hardware setup
- Components of a DAQ device
- Connection types

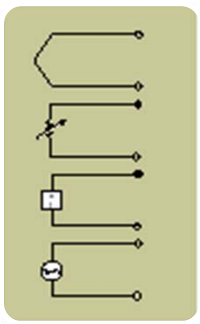


Why Use NI Hardware?

- Tight integration between hardware and software
- High-speed Application-Specific Integrated Circuits (ASIC)
- Extensive price/performance offerings on a wide variety of platforms
- Widest range of possible measurement types
- NIST-traceable calibration for accurate repeatable measurements

Data Acquisition Hardware

DAQ Hardware turns your PC into a measurement and automation system



Sensors
- Any type

Signal Connector
- Direct Connection
- Terminal Block

DAQ Device
- PCI/PXI
- PCIe/PXIe
- USB

Computer
- Windows
- Linux
- Mac (DAQmx Base)

Signal Connector

Route your signal to specific lines on your DAQ device

PCI/PCIe

- Requires
  - Terminal block for connections to the sensors
  - Cable to connect the DAQ device and terminal block

USB

- Signals connect directly to the DAQ device
- Often has sensor-specific connector
  - BNC
  - RJ-50
  - Dsub

BNC-2120 Shielded Connector Block

- Quadrature Encoder
- 8 LEDs for Digital I/O
- Counter I/O
- Function Generator
- Function Generator Frequency and Amplitude Control
- Temperature Sensor
- Analog Input
- Analog Output

DAQ Device

DAQ devices connect to the bus of your computer

Most DAQ devices have:
- Analog Input
- Analog Output
- Digital I/O
- Counters

Specialty devices exist for specific applications
- High speed digital I/O
- High speed waveform generation
- Dynamic Signal Acquisition (vibration, sonar)

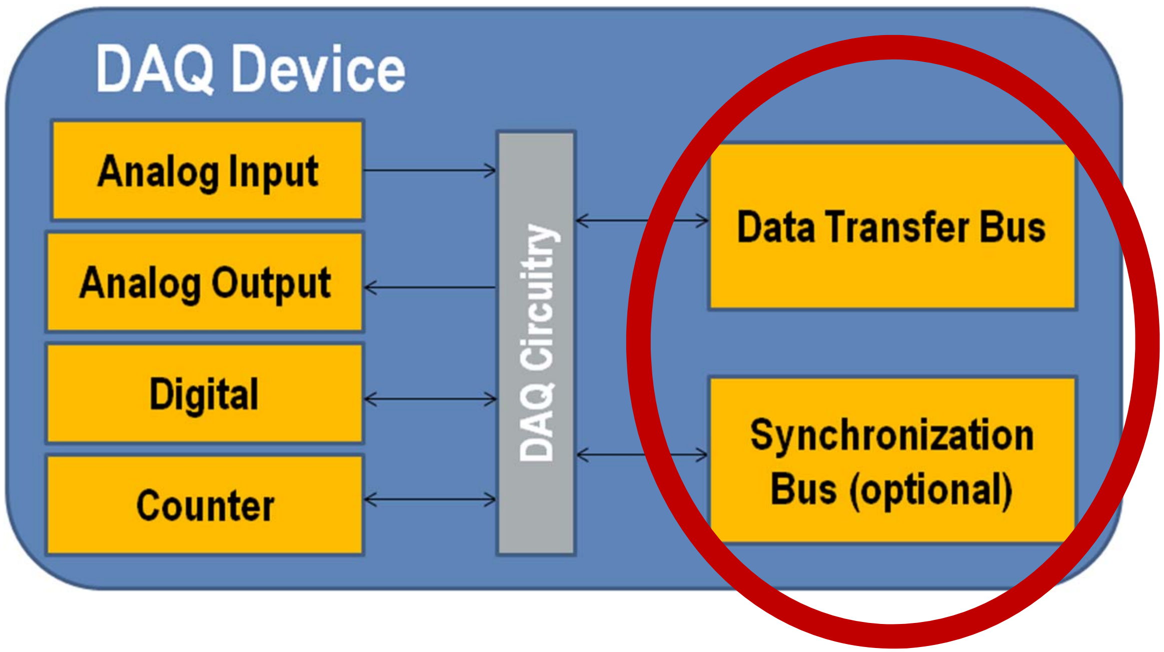
B. Components of a DAQ Device

Data Transfer Bus

- Connects the DAQ device to the computer
- Can be a variety of bus structures
  - USB, PCI. PCI Express, PXI, PXI Express

Synchronization Bus

- Used to synchronize multiple DAQ devices
- Allows sharing of timing and trigger signals between devices
- Not available on USB devices

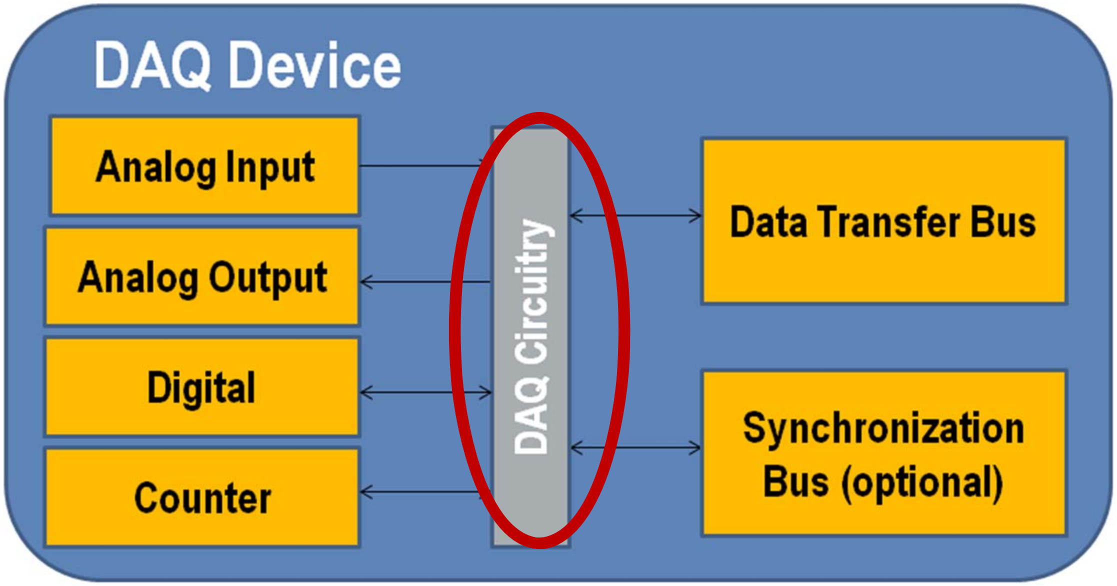


Components of a DAQ Device

DAQ Circuitry

Contains all circuitry necessary for completing a DAQ task

- Clock and timing circuitry
- On-board FIFOs
- Signal routing
- Precision rails for calibration



Components of a DAQ Device

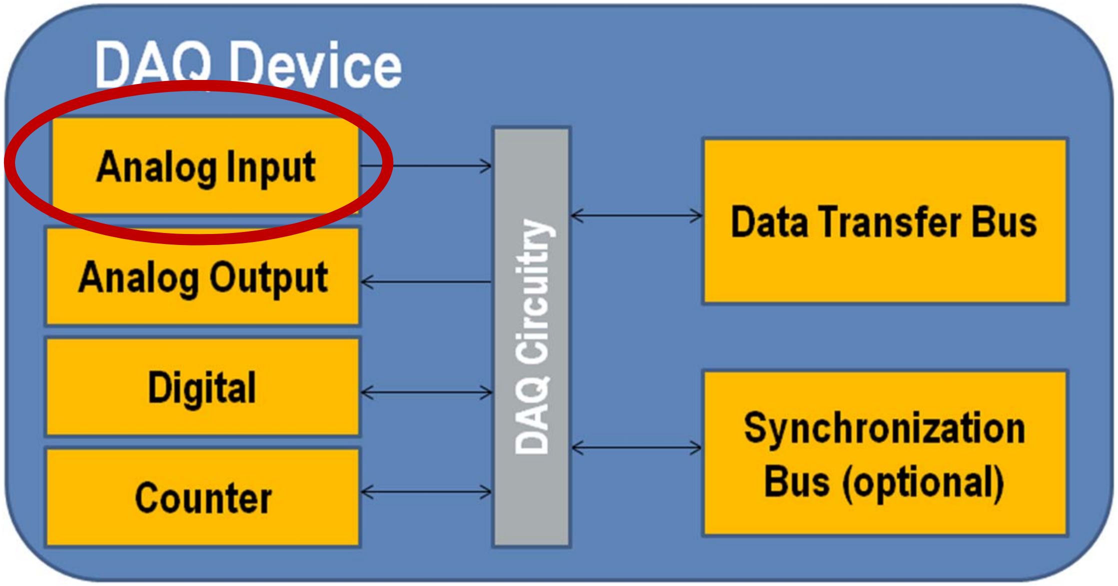
Analog Input Circuitry

Some signal conditioning

- Multiplexing of signals
  - Switch that has multiple input channels but only lets one channel at a time through to the instrumentation amplifier
- Instrumentation Amplifier
  - Either amplifies or attenuates your signal

Analog-to-Digital Converter (ADC)

- Converts an analog signal to a digital number

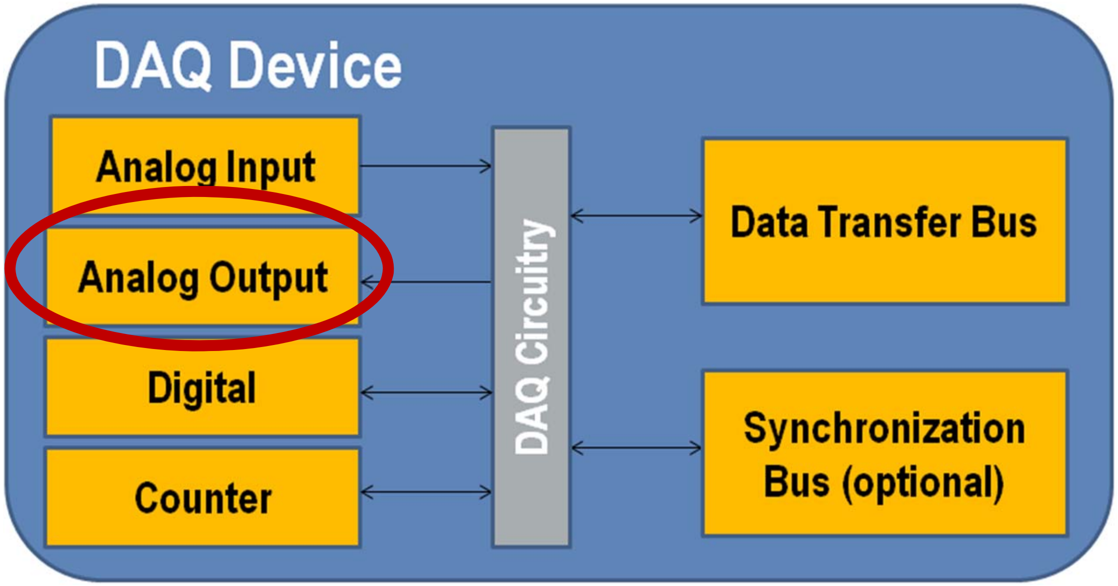


Components of a DAQ Device

Analog Output Circuitry

Digital-to-Analog Converter (DAC)

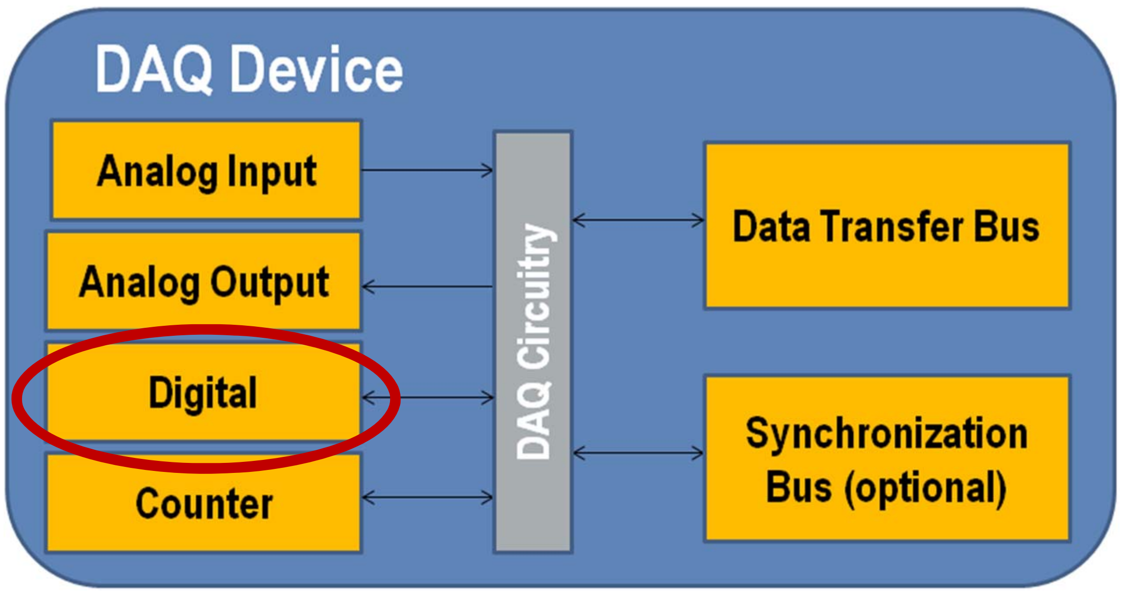
- Converts digital number to analog signal
- Usually one DAC per channel



Components of a DAQ Device

Digital I/O Circuitry

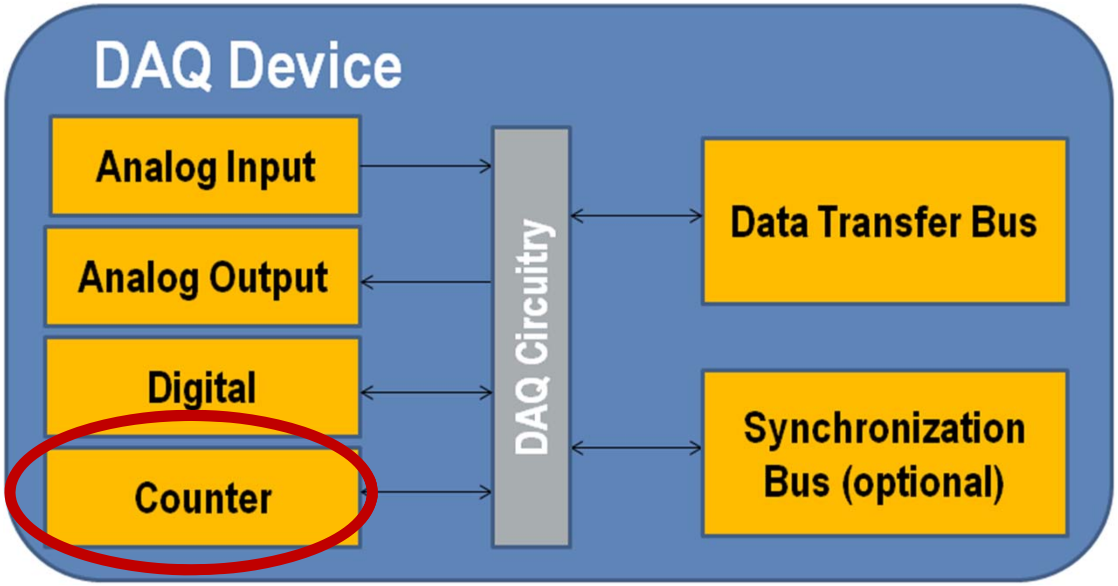
- Can input or output digital signals



Components of a DAQ Device

Counter Circuitry

- Can input or output digital signals
- Suitable for measuring rate
  - Built in timing signals
  - Counting functionality



C. Choosing Appropriate DAQ Hardware

- Bus Considerations
- Signal Considerations
- Accuracy Considerations

Bus Considerations

- How much data will I be streaming across this bus?
  - Bus bandwidth
- What are my single-point I/O requirements?
  - Bus latency and determinism
- Do I need to synchronize multiple devices?
  - Bus synchronization options
- How portable should this system be?
- How far will my measurements be from my computer?

Bus Considerations

Bus	Waveform Streaming	Single-Point I/O	Multi-Device Synchronization	Portability	Distributed I/O
PCI	132 MB/s (shared)	Best	Better	Good	Good
PCI Express	250 MB/s (per lane)	Best	Better	Good	Good
PXI	132 MB/s (shared)	Best	Best	Better	Better
PXI Express	250 MB/s (per lane)	Best	Best	Better	Better
USB	60 MB/s	Better	Good	Best	Better
Ethernet	12.5 MB/s	Good	Good	Best	Best
Wireless	6.75 MB/s	Good	Good	Best	Best

Signal Considerations

- How many channels?
  - Choose DAQ device(s) with enough channels
- How quickly do you need to acquire/generate samples of the signal?
  - Choose DAQ device with fast enough sampling rate
- What are the expected minimum and maximum measurements?
  - Choose DAQ device with appropriate range

Signal Considerations

- What is the smallest change in your signal that you need to detect?
  - Choose DAQ device with a small enough code width
  - To calculate the code width, you must know:
    - Resolution
    - Device input range

Calculating Code Width – Resolution

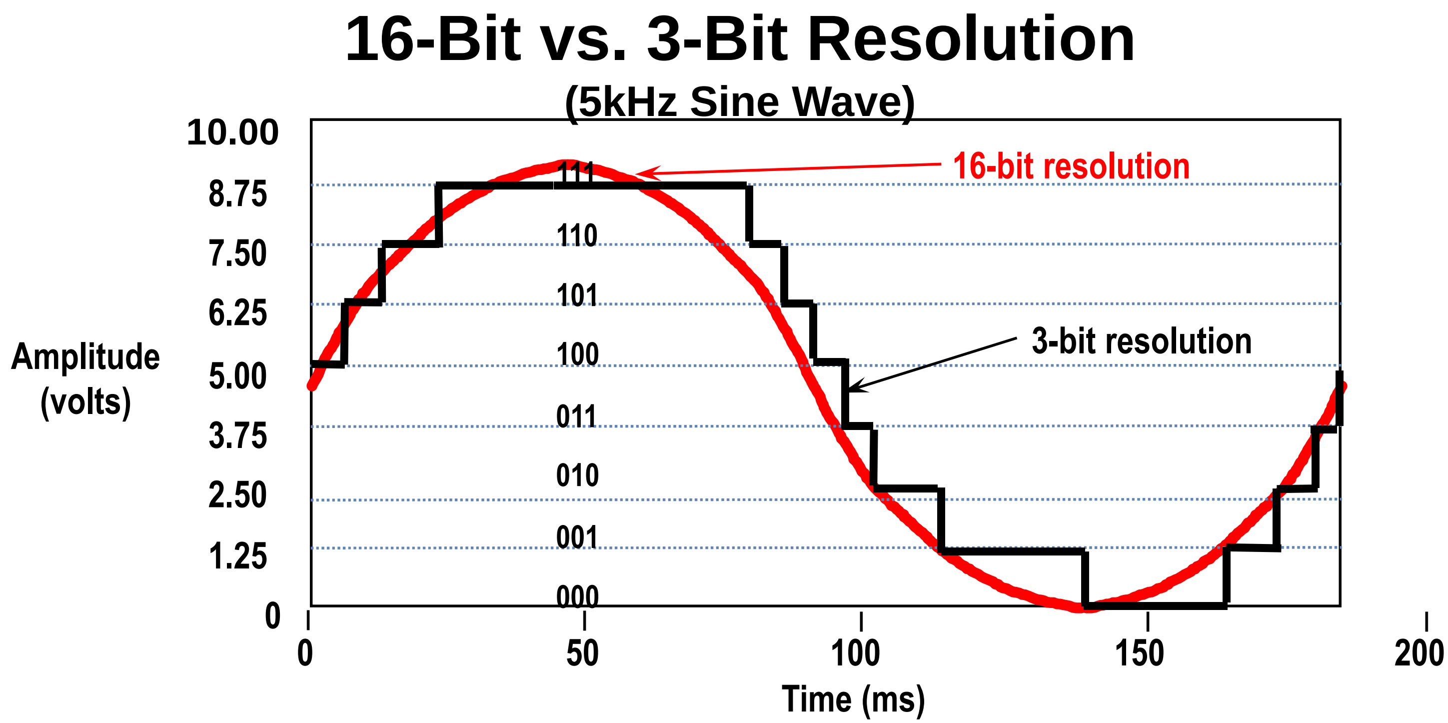
- Resolution
  - Number of bits the ADC uses to represent a signal
- Resolution determines how many different voltage changes can be measured
- Example: 16-bit resolution

$$\# \text{ of levels} = 2^{\text{resolution}} = 2^{16} = 65{,}536 \text{ levels}$$

- Larger resolution = more precise representation of your signal

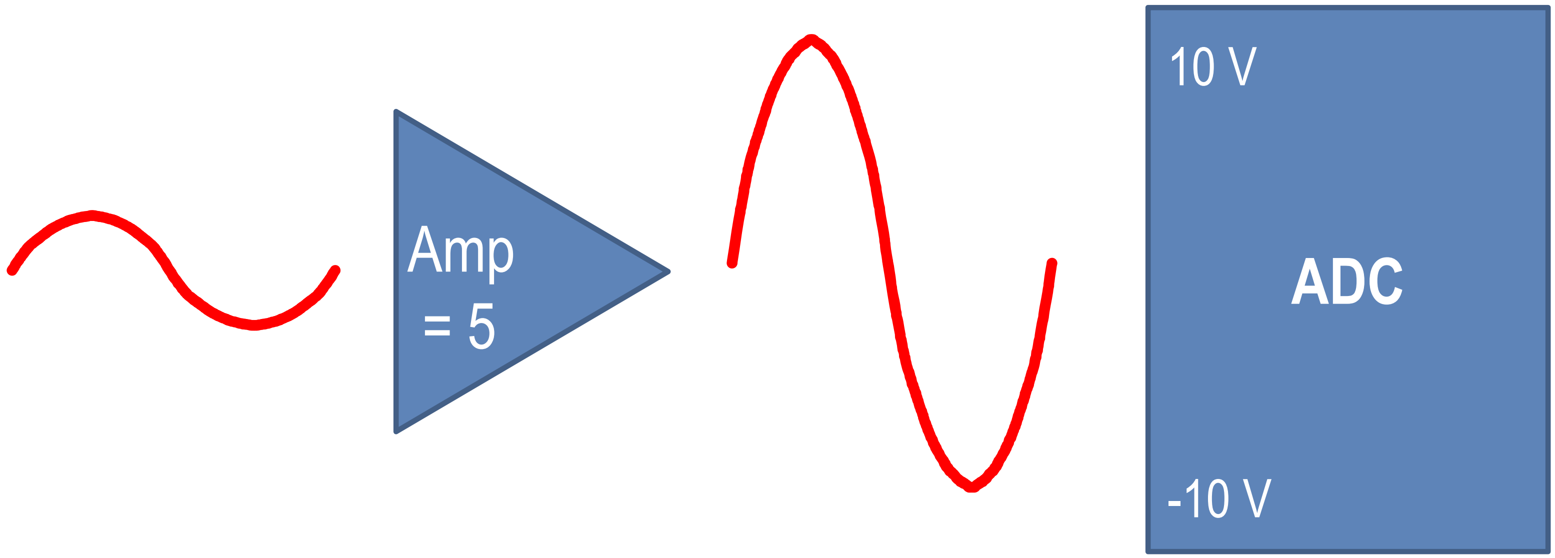
Calculating Code Width – Resolution Example

- 3-bit resolution can represent 8 voltage levels
- 16-bit resolution can represent 65,536 voltage levels



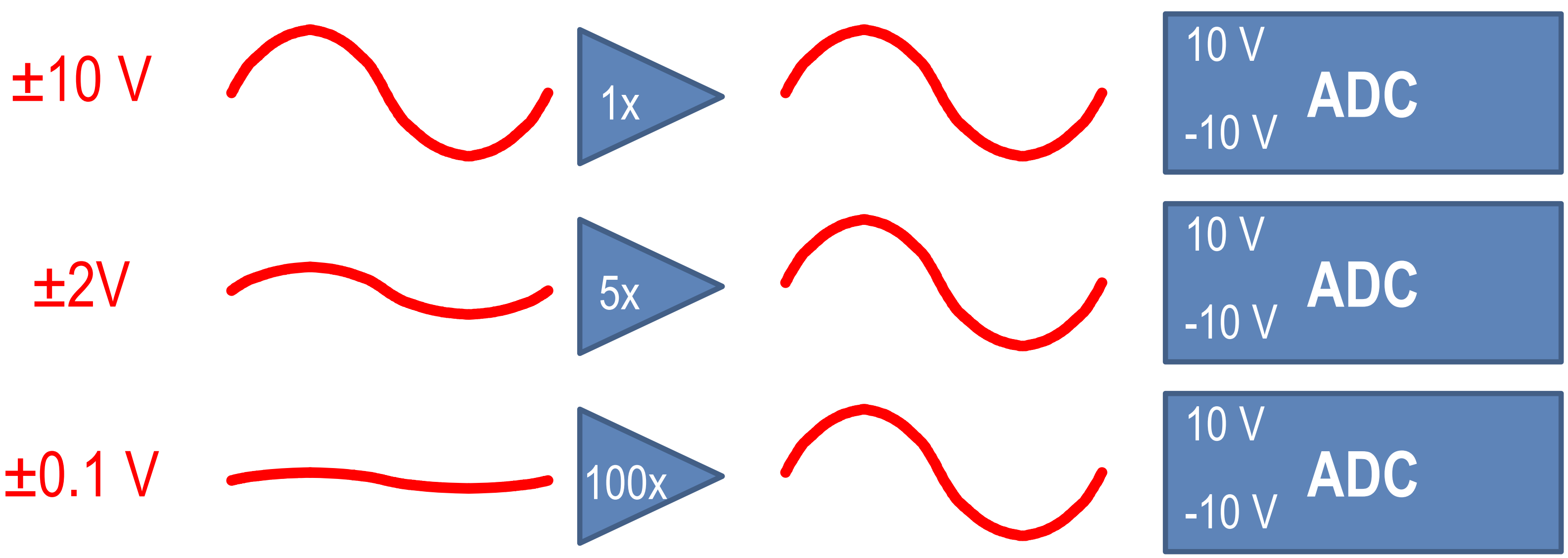
Calculating Code Width – Amplification and Device Input Range

- DAQ devices have a built-in amplifier
  - Amplifies the signal to better fit the range of the ADC
  - Better utilizes the ADC resolution



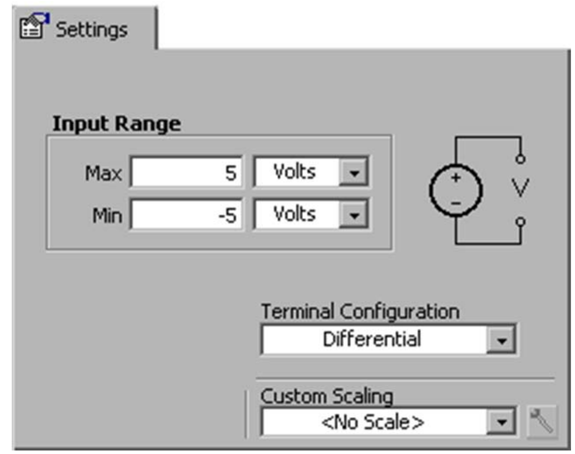
Calculating Code Width – Amplification and Device Input Range

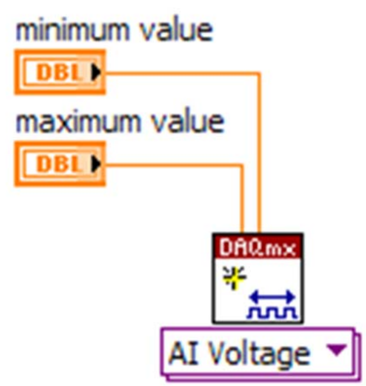
- The amplification gains available on the DAQ device determine the device input ranges available
  - Example device input ranges include ±10, ±5, ±2, ±1, ±0.5, ±0.2, ±0.1 V



Calculating Code Width – Amplification and Device Input Range

- You do not set the amplification gains or device input ranges directly
- You set your minimum and maximum expected values in software
  - DAQ device automatically chooses which device input range to use based on your min/max settings
    - Min=-3V, Max=3.5V → ±5V device input range
    - Min=-9V, Max=8V → ±10V device input range
- Setting proper min/max
  - More precise representation of your signal
  - Utilizes all of your available resolution





DAQmx Create Virtual Channel.vi

Calculating Code Width

Code width is the smallest change in the signal that your system can detect (determined by resolution and device input range)

$$\text{code width} = \frac{\text{Device input range}}{2^{\text{resolution}}}$$

Smaller Code Width = more precise representation of your signal

Example: 16-bit device, device input range = ±10 V

$$\frac{\text{Device input range}}{2^{\text{resolution}}} = \frac{10-(-10)}{2^{16}} = 305\ \mu V$$

Use smaller device input range:
$$\frac{5-(-5)}{2^{16}} = 153\ \mu V$$

Use smaller device input range and use device with higher resolution:
$$\frac{5-(-5)}{2^{18}} = 38\ \mu V$$

Accuracy Considerations

- How close to the true value does your measurement need to be?
  - Make sure your DAQ device has an acceptable absolute accuracy
  - Absolute accuracy defines the overall uncertainty of your measurement
- Accuracy considerations
  - Code Width ≠ Accuracy
  - Sources of error affecting accuracy
    - Gain errors and offset errors from amplifier and ADC
    - Noise in the system

Accuracy Considerations

- Use the specifications manual of the DAQ device
  - Lists the absolute accuracy for each device input range
  - Lists absolute accuracy equation and numbers for each component if you want to calculate it yourself

- Example AI Absolute Accuracy table
  - When this DAQ device is using a **±2 V** device input range, the absolute accuracy of the measurement will be within **±410 μV** of the true value

Nominal Range		Absolute Accuracy at Full Scale[1] (μV)
Positive Full Scale	Negative Full Scale	
10	–10	1,920
5	–5	1,010
2	–2	410
1	–1	220
0.5	–0.5	130
0.2	–0.2	74
0.1	–0.1	52

Exercise 2-1: Device Input Range, Resolution, Code Width, and Accuracy

To determine the optimal configuration for a data acquisition measurement system.

GOAL

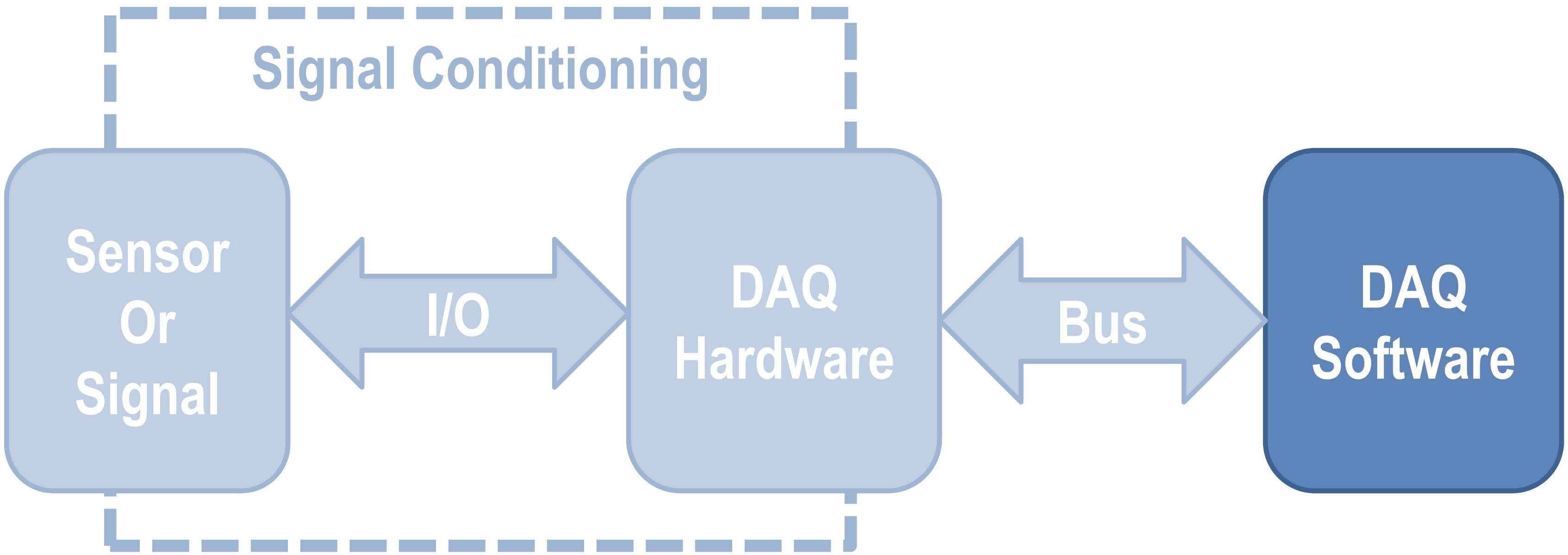
Exercise 2-1: Device Input Range, Resolution, Code Width, and Accuracy

- If you increase the resolution of your DAQ device, what happens to the code width?
- If you increase the device input range, what happens to code width?

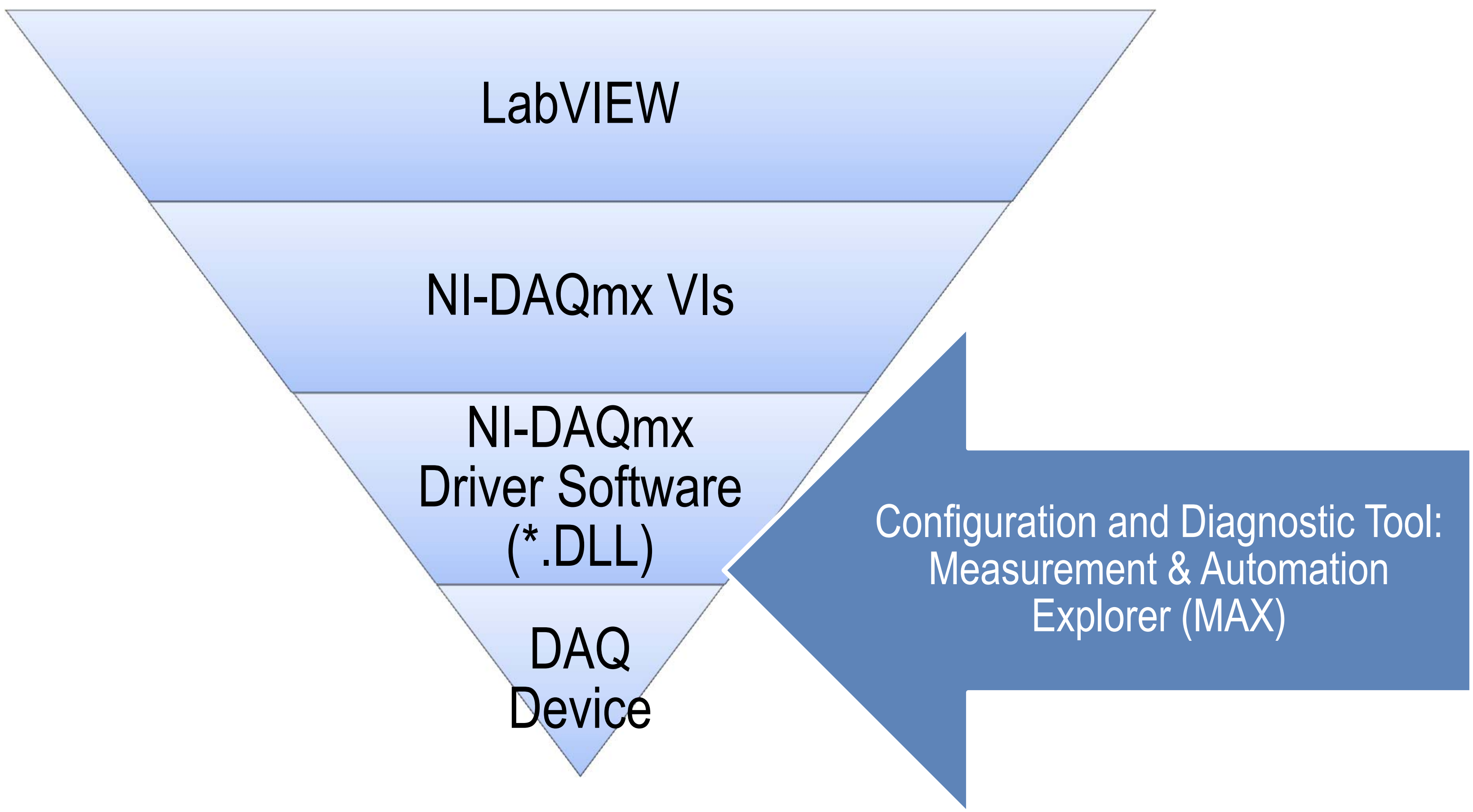
DISCUSSION

D. DAQ Software Overview

- NI-DAQmx Software Architecture
- NI-DAQmx Overview
- Measurement & Automation Explorer (MAX) Overview
- DAQ Assistant



NI-DAQmx Software Architecture



What is NI-DAQmx?

- Driver-level software
  - DLL that makes direct calls to your DAQ device
- NI-DAQmx does not support 3rd-party data acquisition devices
- Supports the following National Instruments software:
  - LabVIEW
  - Measurement Studio
  - Signal Express
  - LabWindows/CVI
  - LabVIEW Real-Time Module

NI-DAQmx Platform Support

Also supports the following 3rd party languages:

- Microsoft Visual Basic .NET
- Microsoft Visual Basic 6.0
- Microsoft Visual C/C++
- Microsoft C# .NET
- ANSI C

Benefits of NI-DAQmx

- DAQ Assistant
- Increased performance – faster single point I/O and multithreading
- Simple and intuitive API
- DAQ property nodes and waveform support
- Similar API for all programming languages
- Run NI-DAQmx programs and Assistant without the hardware!!
- Reduces development time with its interactive features

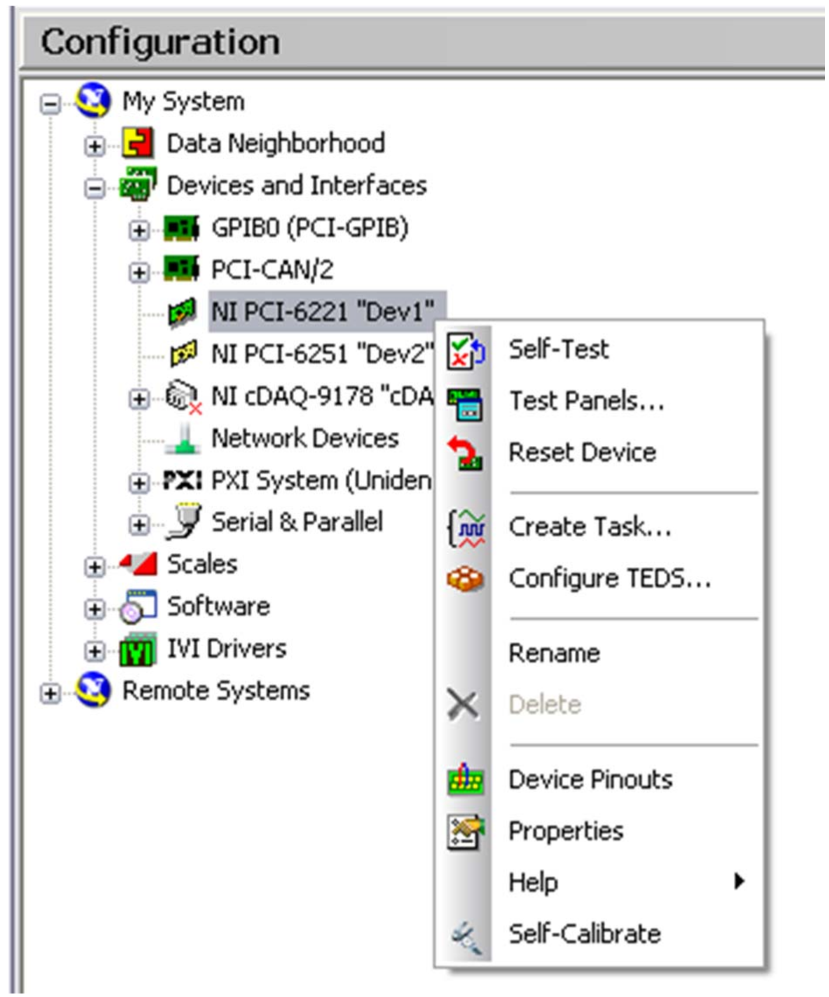
What is MAX?

- MAX stands for Measurement & Automation Explorer
- MAX provides access to all your National Instruments DAQ, GPIB, IMAQ, IVI, Motion, VISA, CAN, Modular Instruments, PXI, and VXI devices
- Used for configuring and testing devices
  - Data Neighborhood
  - Devices and Interfaces
  - Historical Data
  - Scales
  - Software
  - VI Logger Tasks
  - IVI Drivers
  - Remote Systems

Measurement
& Automation

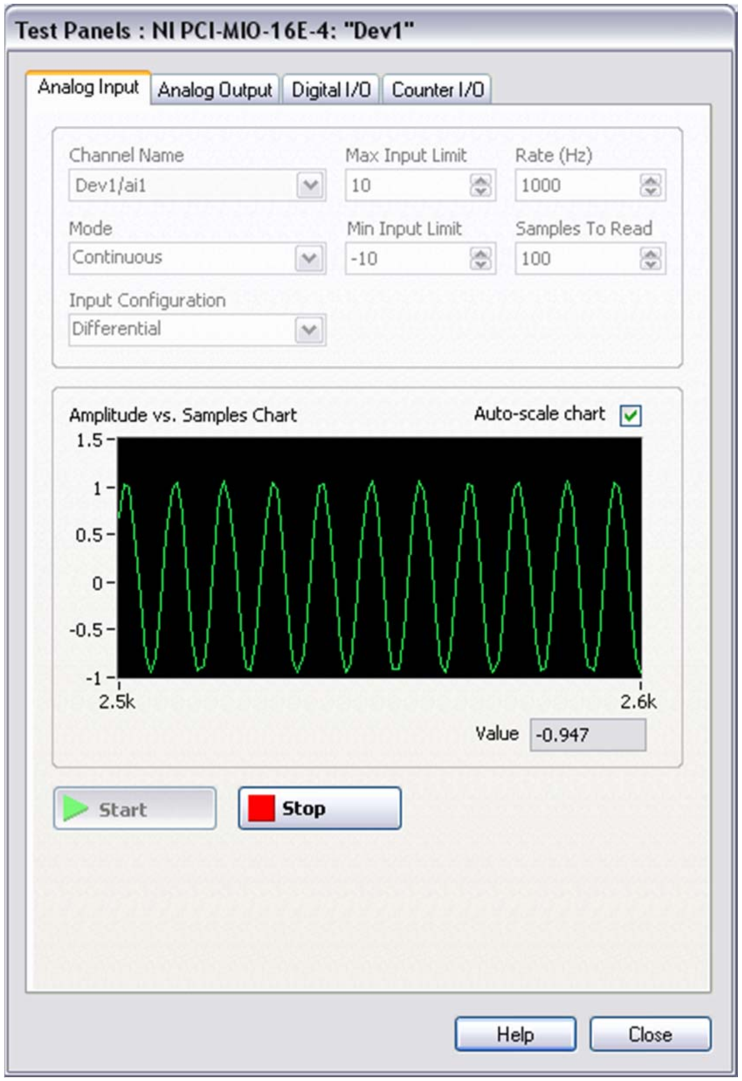
Devices and Interfaces

- Shows currently installed and detected National Instruments hardware
- Includes utilities for configuring and testing your DAQ devices
  - Self-Test
  - Test Panels
  - Reset
  - Properties
  - Self-Calibrate



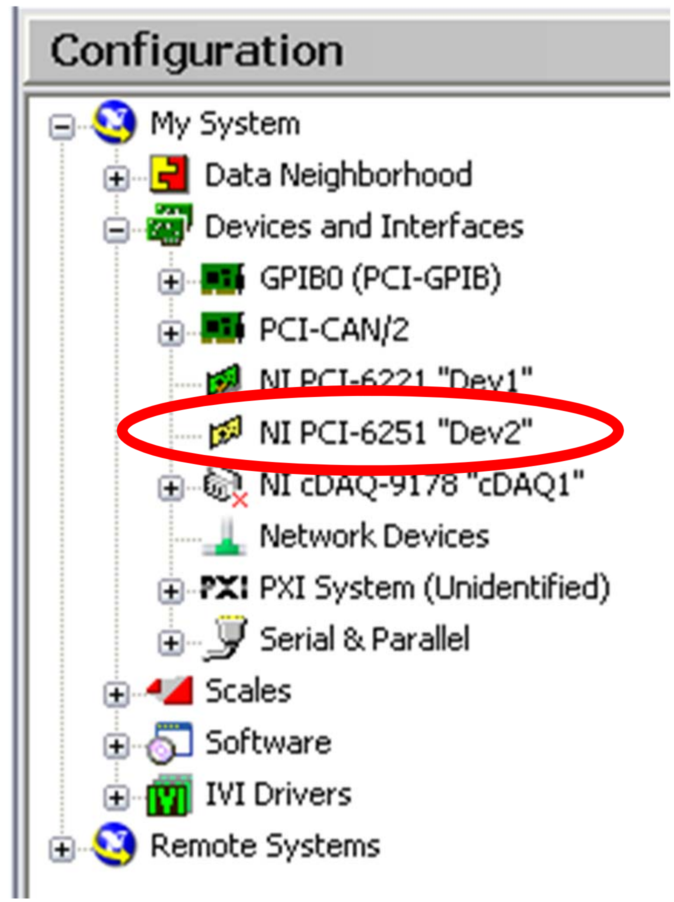
Test Panels

- Utility for testing
  - Analog Input
  - Analog Output
  - Digital I/O
  - Counter I/O
- Great tool for troubleshooting



NI-DAQmx Simulated Devices

- Run NI-DAQmx programs and Assistant without the hardware!
- Assistant and programs run just like on a real device with some exceptions:
  - Timing and triggering are instantaneous
  - Reads return simulated data (for AI, data is a sine wave with some noise)
- Most DAQmx devices are supported (DAQmx plug-in devices, cDAQ, and more)



Exercise 2-2: Using Measurement & Automation Explorer

To become familiar with the Devices and Interfaces section of MAX and to explore the test panel functionality.

GOAL

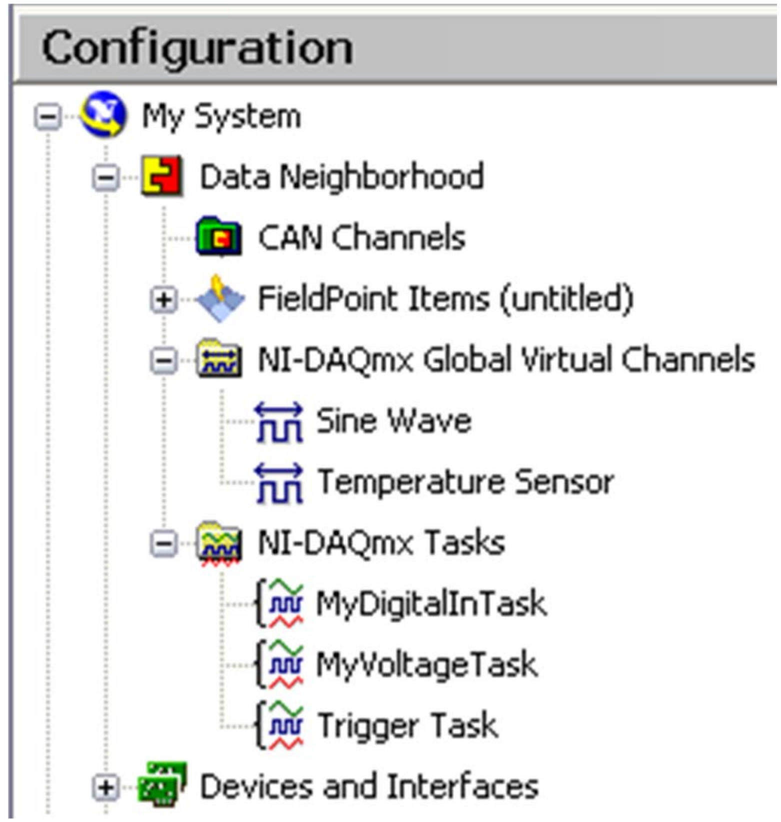
Exercise 2-2: Using Measurement & Automation Explorer

- How can you verify if the analog output channels on your DAQ device are outputting the correct voltages?

DISCUSSION

Data Neighborhood

- Provides access to DAQ Assistant
- Shows configured tasks and channels
- Includes utilities for testing and reconfiguring tasks and channels



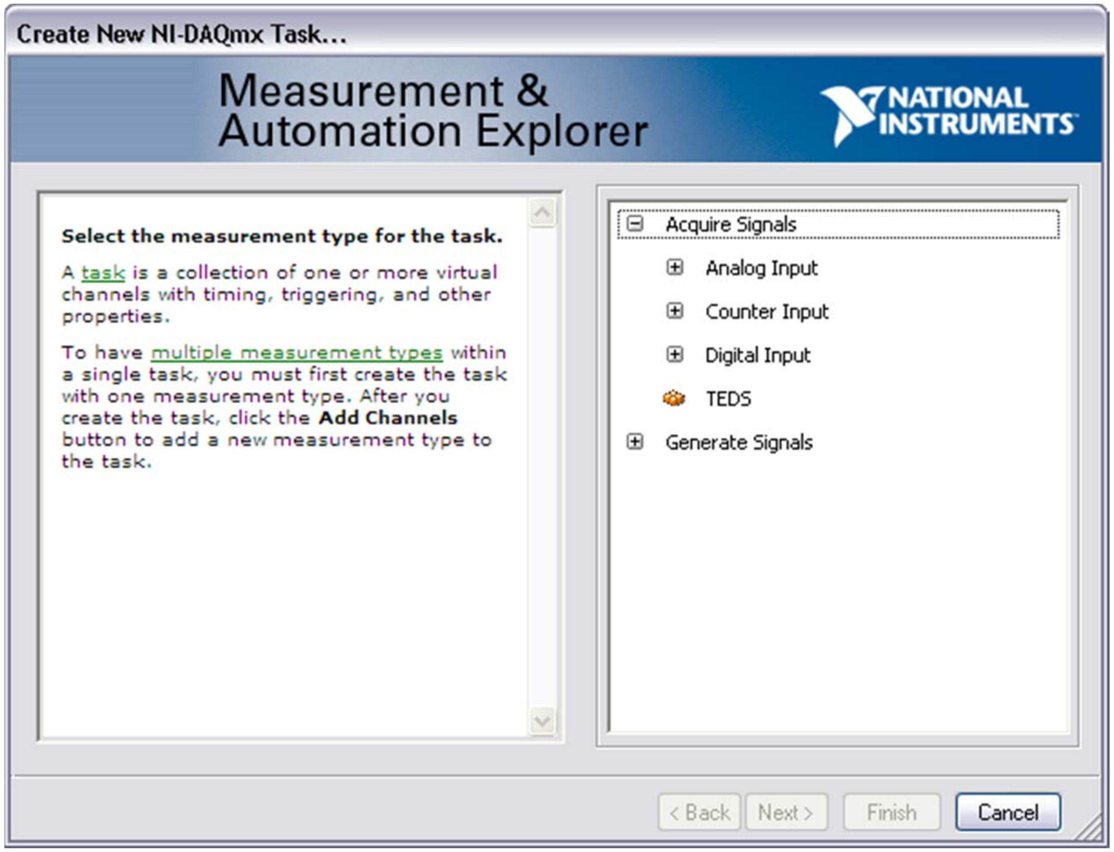
DAQ Assistant – Channels

Interface to create channels for:

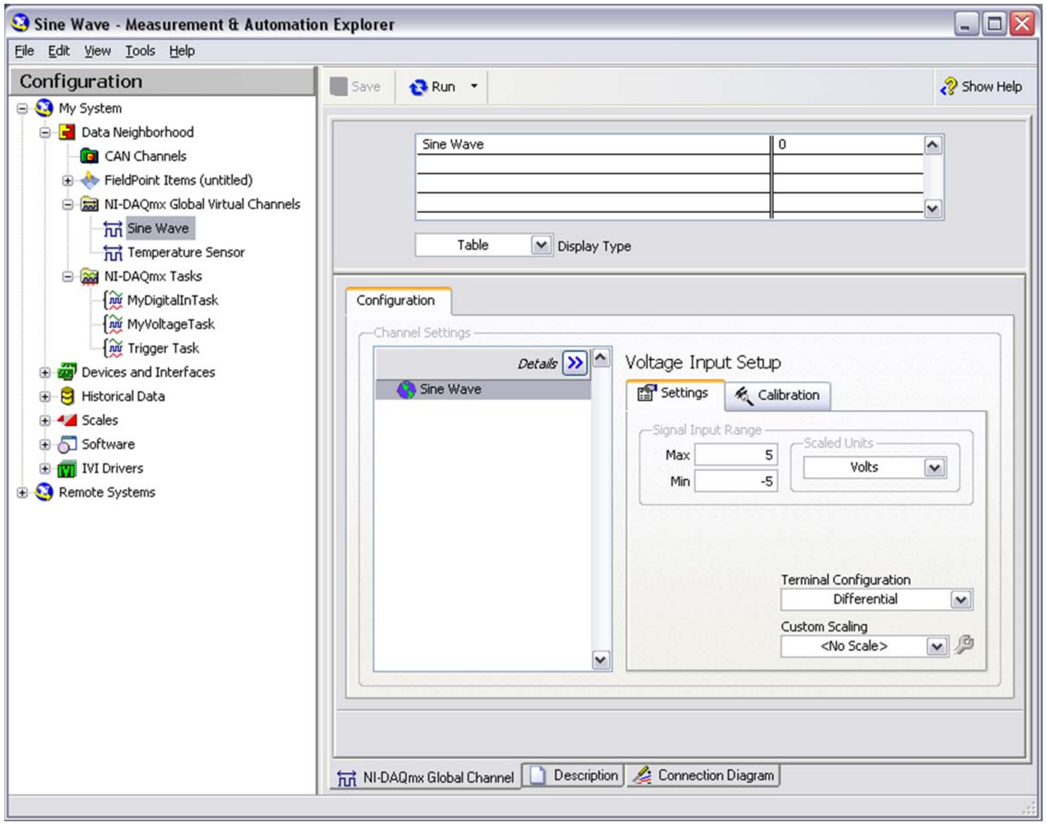
- Analog Input
- Analog Output
- Counter Input
- Counter Output
- Digital I/O
- TEDS

Each channel has:

- Measurement type
- Sensor/signal type
- Name



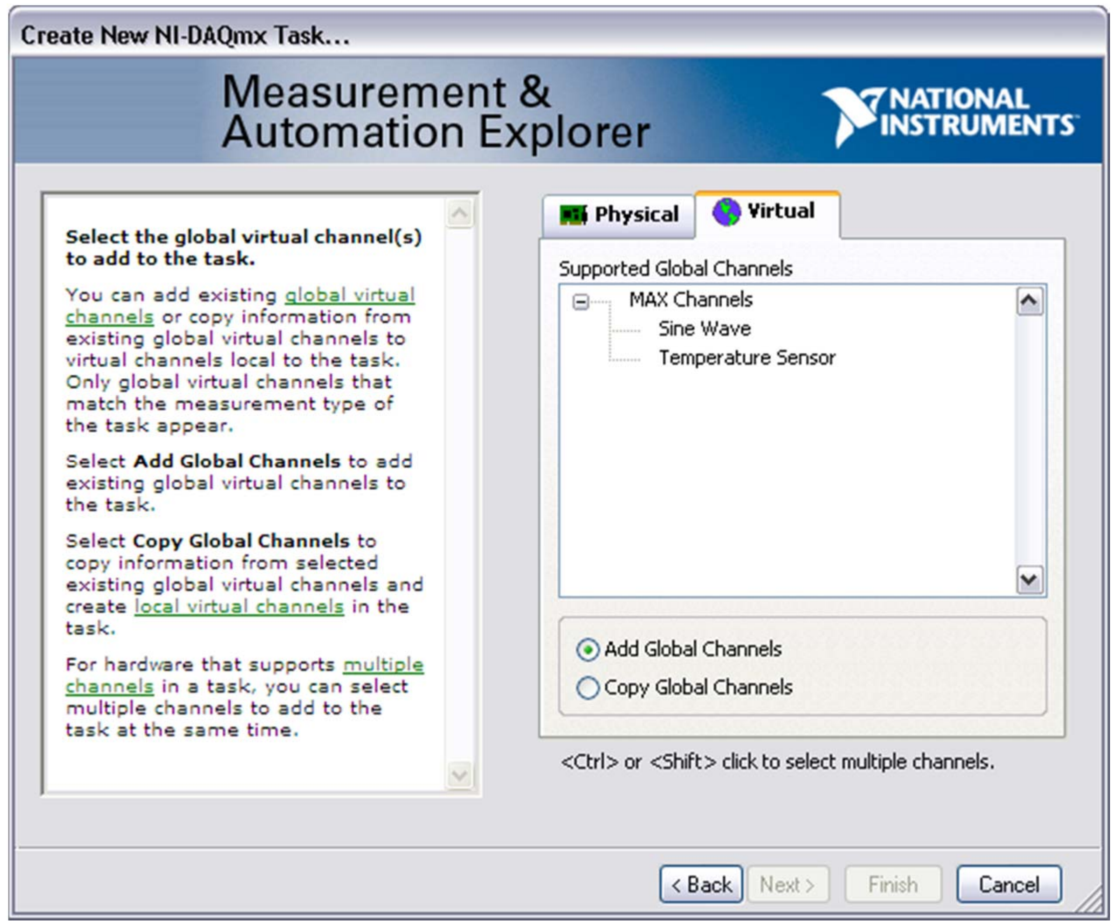
Channel Configuration



- Configure settings:
  - Min/max (determines which amplification and device input range the DAQ device uses)
  - Terminal configuration
  - Custom Scaling
- Launch Test Panel
- Connection diagram

DAQ Assistant – Tasks

- Task: A collection of channels with homogeneous timing and triggering
- Use new or existing channels

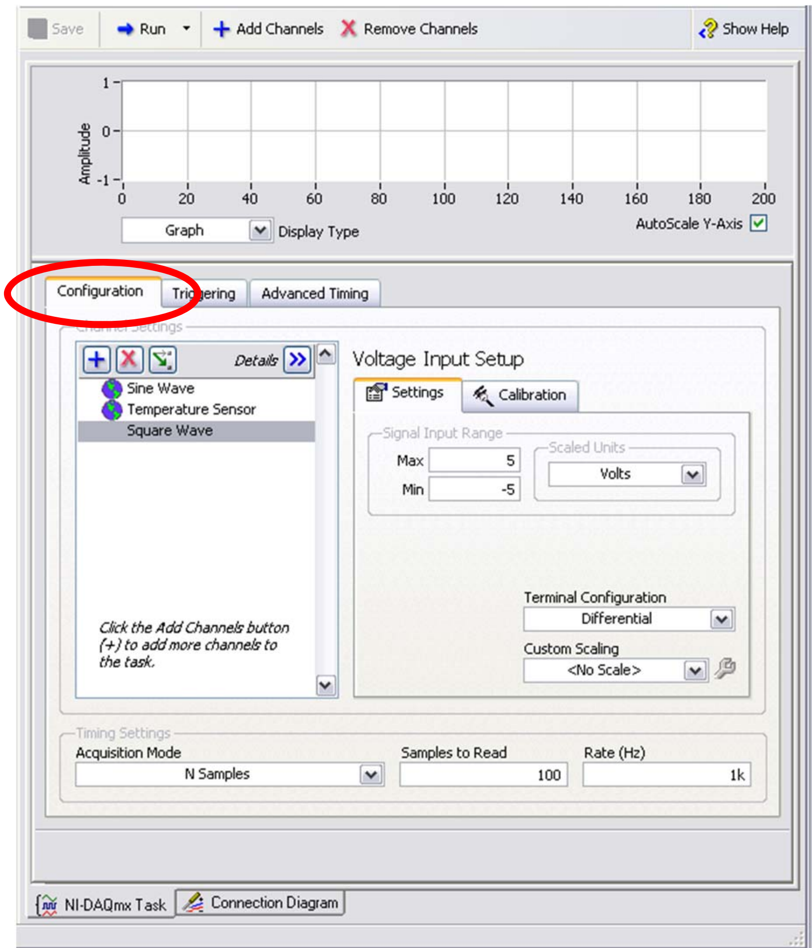


Scope of Channels within a Task

- Local Channels: Can only be used in that particular task
- Global Channels: Can be used in multiple tasks and referenced outside the context of a task

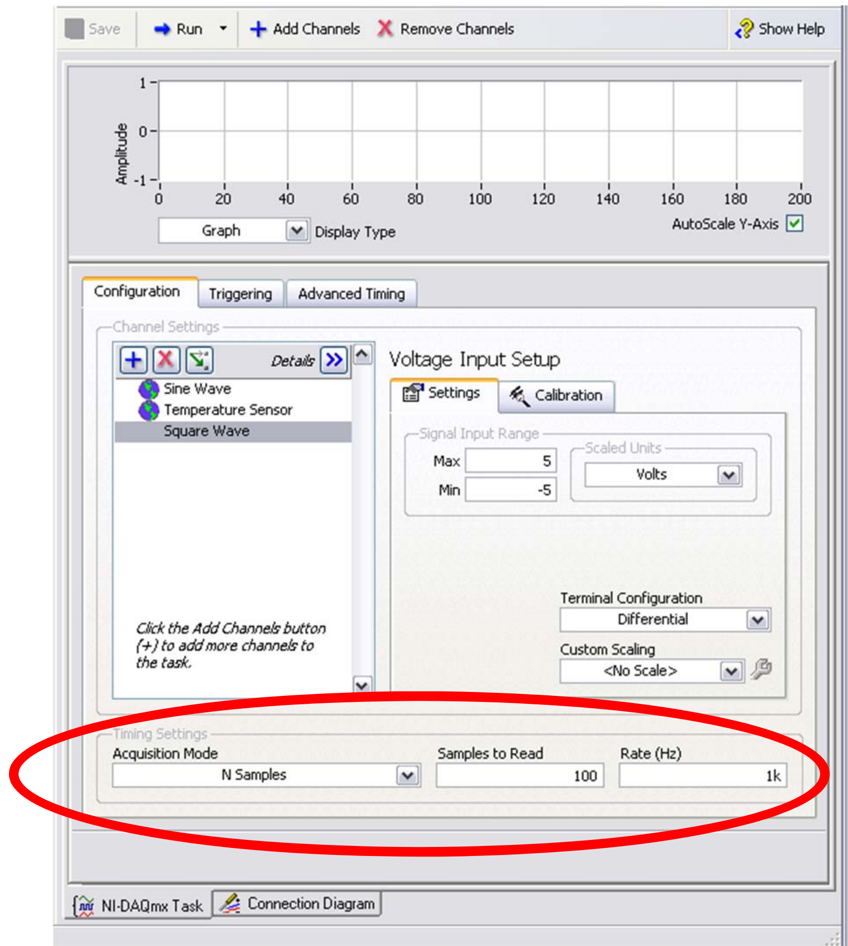
Task Configuration

- Configure settings for physical channels in the task
- Configure settings for global channels in **Data Neighborhood»NI-DAQmx Global Channels**



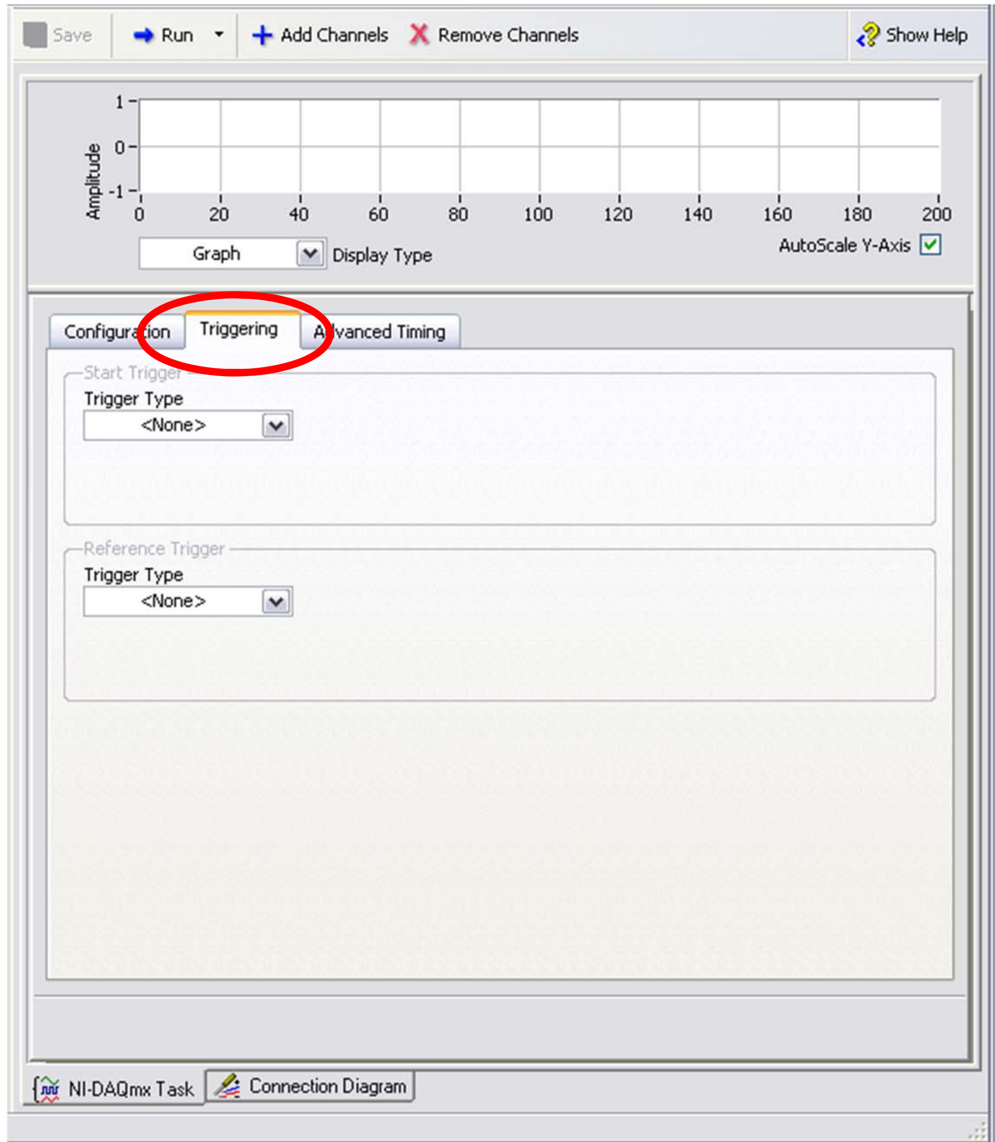
Task Timing

- Configure task timing – single sample (On Demand), single sample (HW Timed), finite, or continuous
- Configure number of samples to read and sampling rate
- Select clock settings



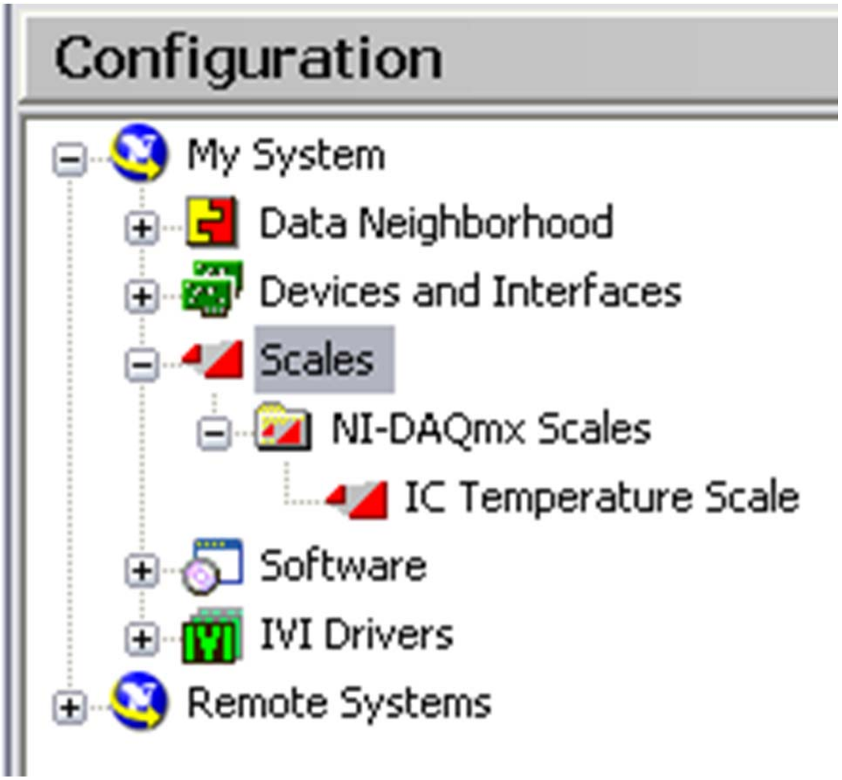
Task Triggering

Configure start and reference triggers



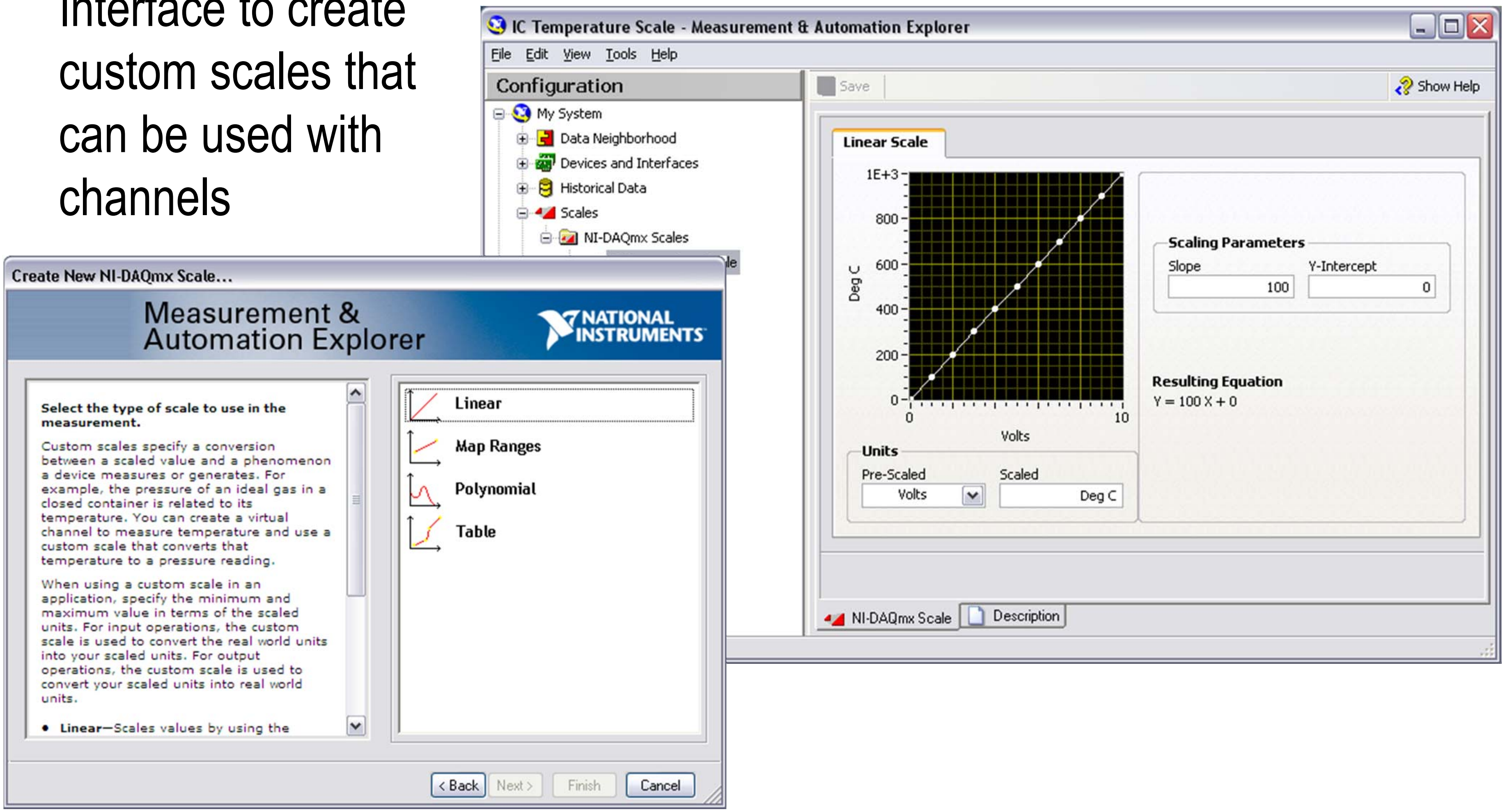
Scales

- Provides access to DAQ Custom Scales Wizard
- Shows configured scales
- Includes utility for viewing and reconfiguring your custom scales



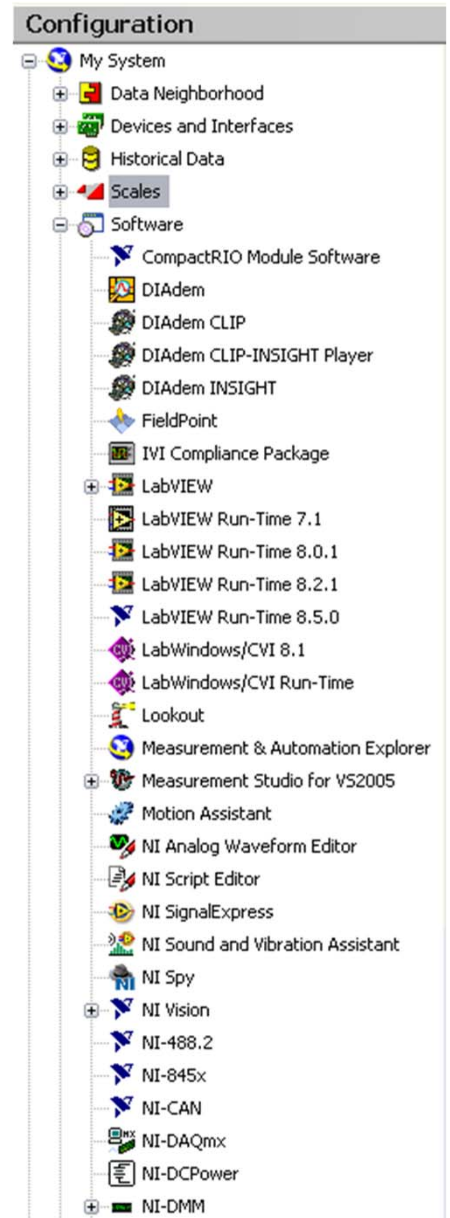
DAQ Custom Scales Wizard

Interface to create custom scales that can be used with channels



Software

- Shows currently installed National Instruments software
- Icon is a shortcut to launch your software
- Includes Software Update Wizard
  - Checks if your NI software is the latest version
  - Links to www.ni.com to download the latest version



Exercise 2-3: DAQ Assistant and Custom Scales Wizard

To create NI-DAQmx channels using the DAQ Assistant and then to create an NI-DAQmx task from these three channels. Also, you will create a custom scale to convert the temperature sensor's voltage to degrees Celsius.

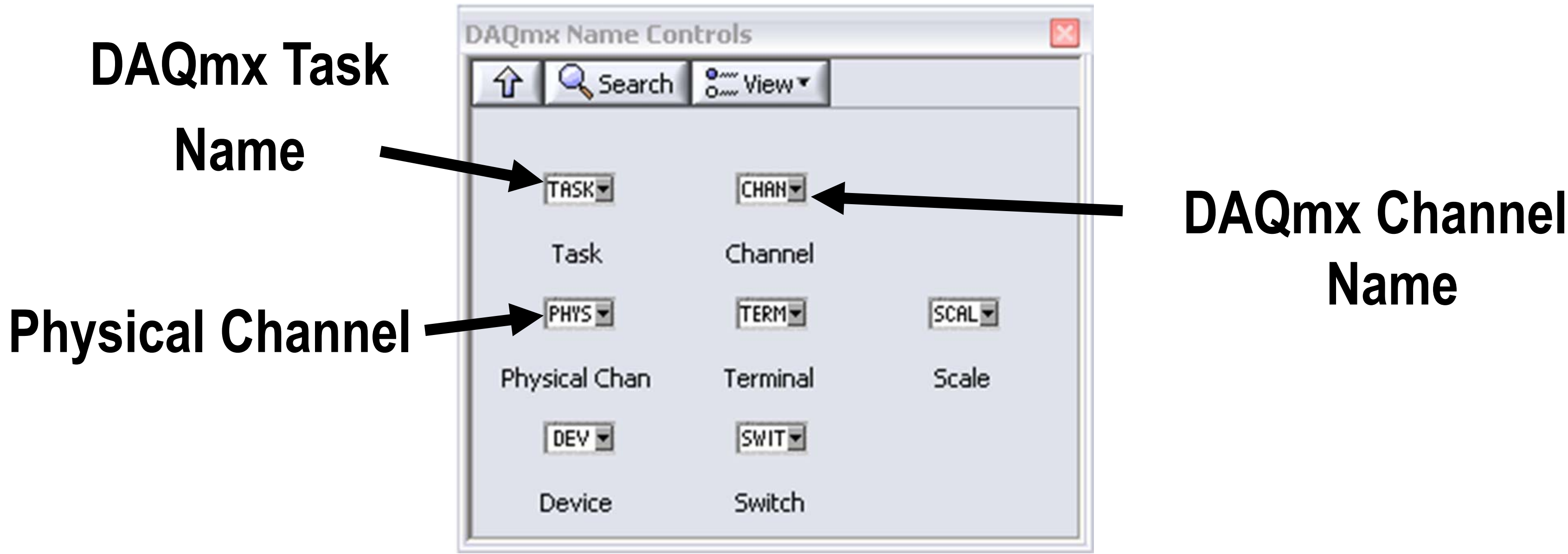
GOAL

Exercise 2-3: DAQ Assistant and Custom Scales Wizard

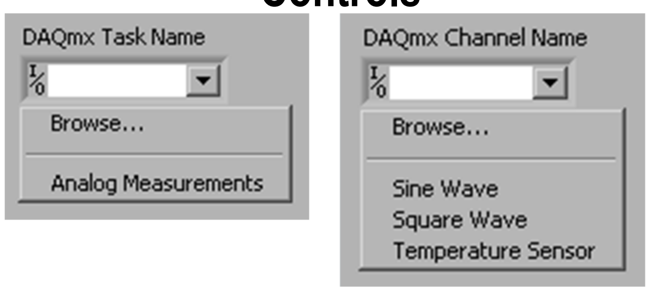
- Can you reuse the local Square Wave virtual channel you created in a different task?

DISCUSSION

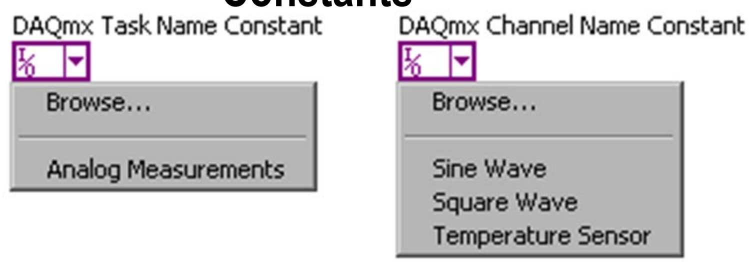
DAQmx Name Controls



Controls

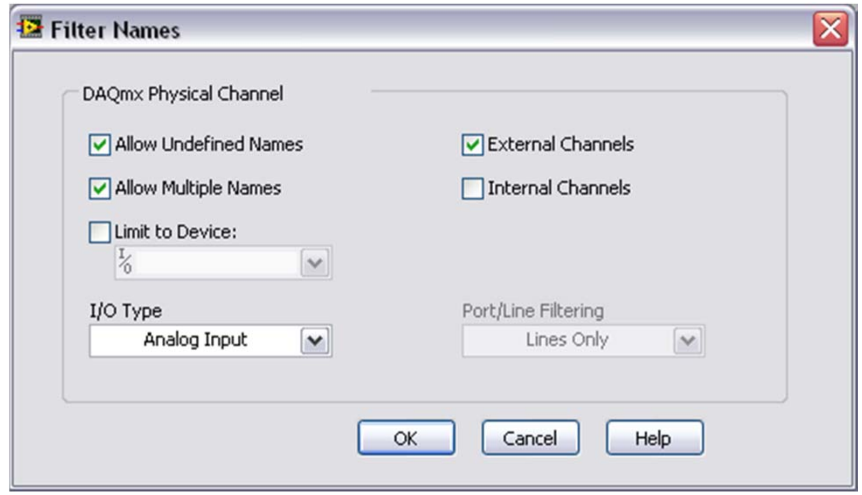


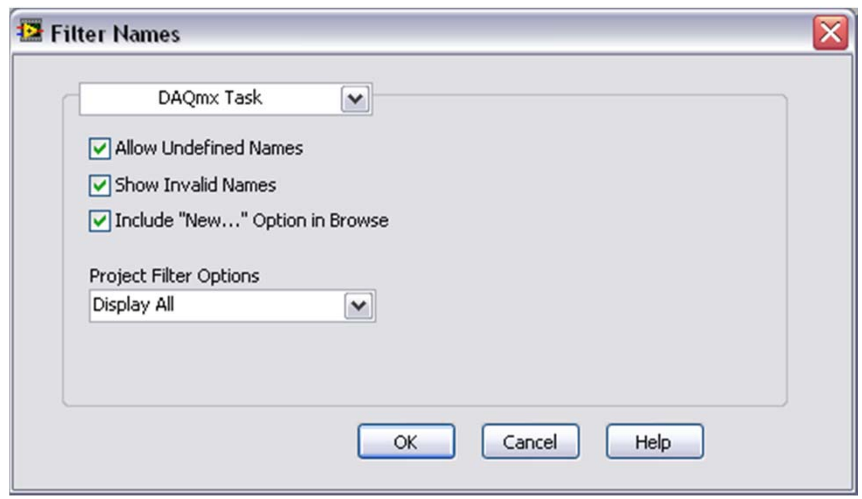
Constants



I/O Name Filtering

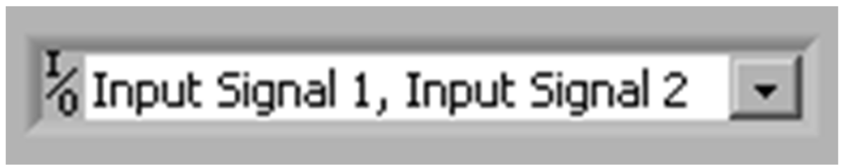
Allows you to set display and filtering options for your channels and tasks





Addressing Multiple Channels

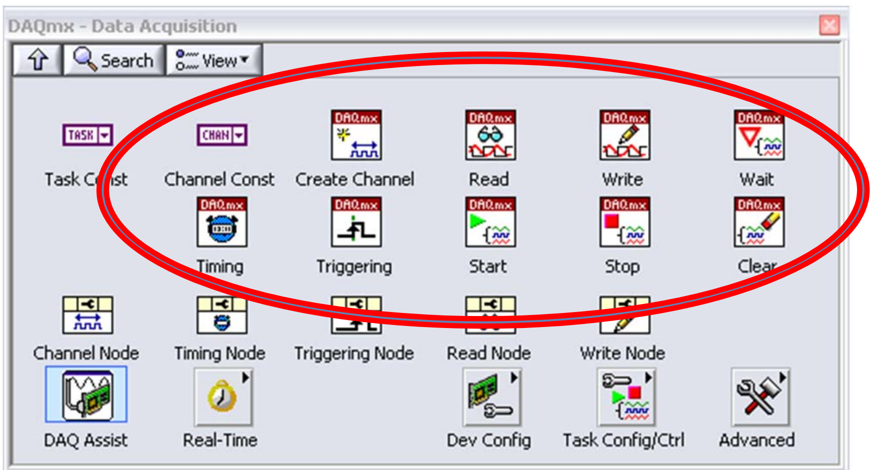
Dissimilar Scales, Ranges, and Terminal Configurations? → Use Multiple Channels



- MAX Channels
  - Separate channel names with a comma when creating task
- Dynamically Created Channels
  - Create each channel separately and add to task

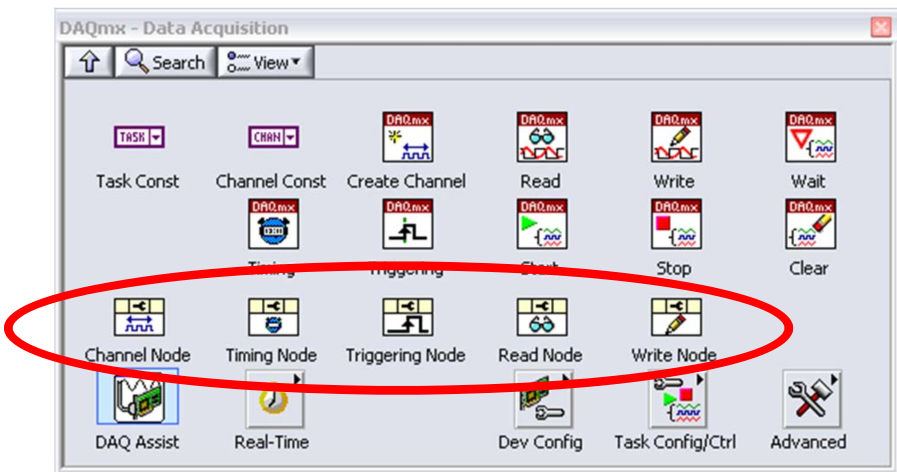
Note: You can only reference multiple channels, not multiple tasks!

E. Overview of NI-DAQmx VIs – Primary Functions



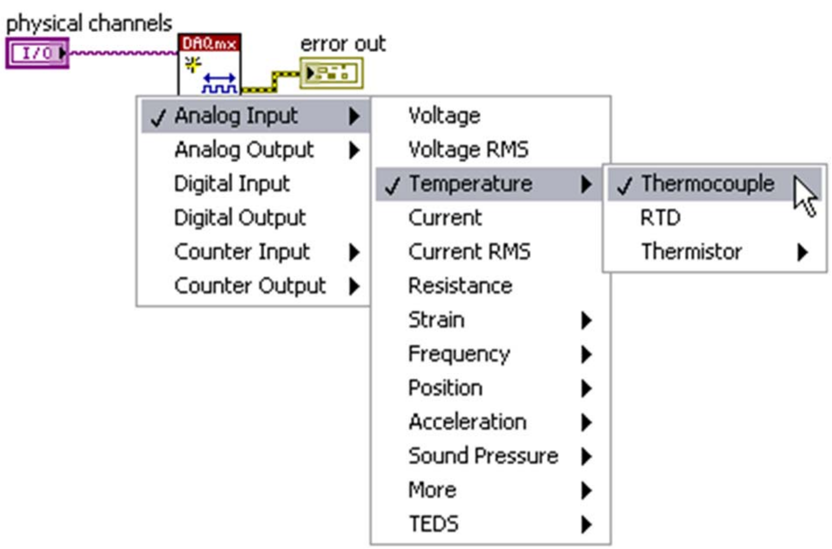
- Create Virtual Channel
- Read
- Write
- Timing
- Trigger
- Task Functions

Overview of NI-DAQmx VIs — Property Nodes



- Property Node – Used to read or write VI and object properties
- Specific property node for
  - Channel
  - Timing
  - Triggering
  - Reading
  - Writing

Create Virtual Channel VI & Channel Property Node

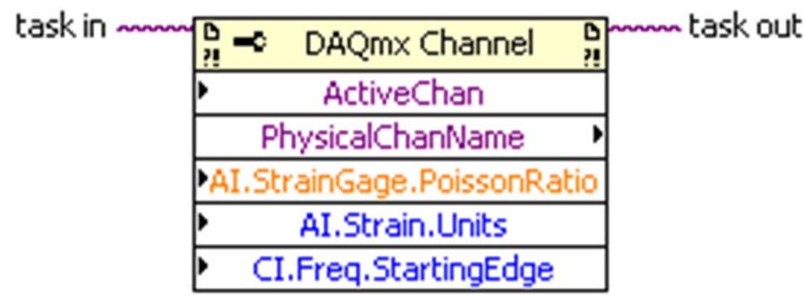


- **Create Virtual Channel VI**
  - Programmatic creation of virtual channel(s)
  - Adds the created channel(s) to a specified task

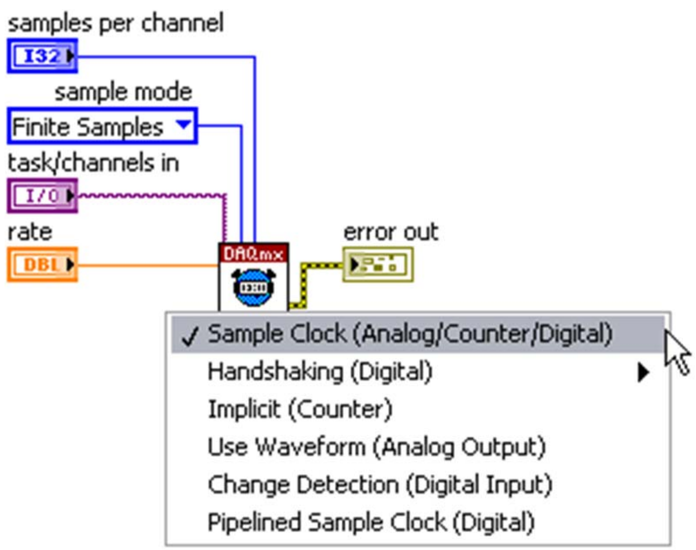
Properties include

- Channel Type
- Physical Channel Name
- Description
- Analog I/O Custom Scale Name
- Digital I/O Number of Lines
- Counter I/O Pulse Duty Cycle

…And many more!



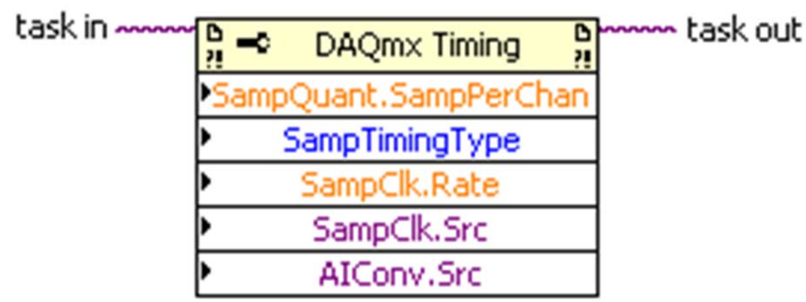
Timing VI & Timing Property Node



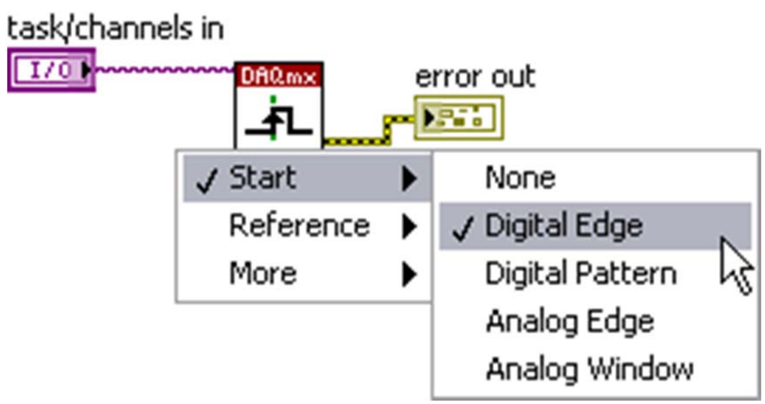
- **Timing VI**
  - Configures sample timing and task duration
  - Creates buffer when needed

- **Properties include**
  - Sample Mode
  - Samples per Channel
  - Sample Timing Type
  - Sample Clock Source
  - Master Timebase Source

  …And many more!



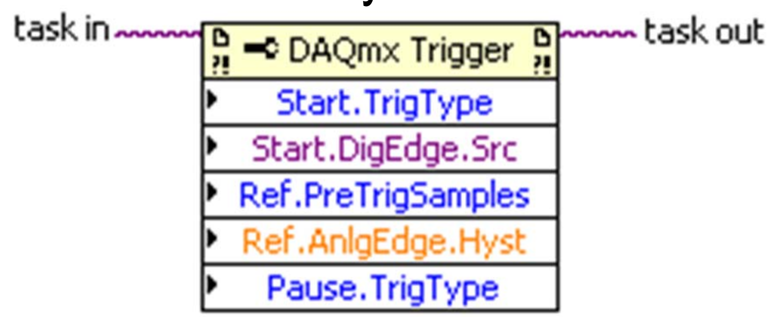
Trigger VI & Trigger Property Node



- **Trigger VI**
  - Configures the task to start or stop on a rising or falling digital edge, analog edge, or analog windows

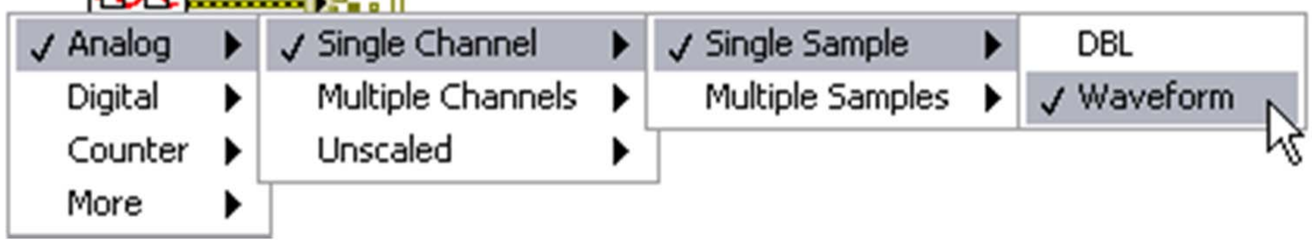
- **Properties include**
  - Start Trigger Type
  - Start Digital Edge Source
  - Start Analog Window Top
  - Reference Pre-Trigger Samples per Channel
  - Reference Analog Edge Slope

  …And many more!



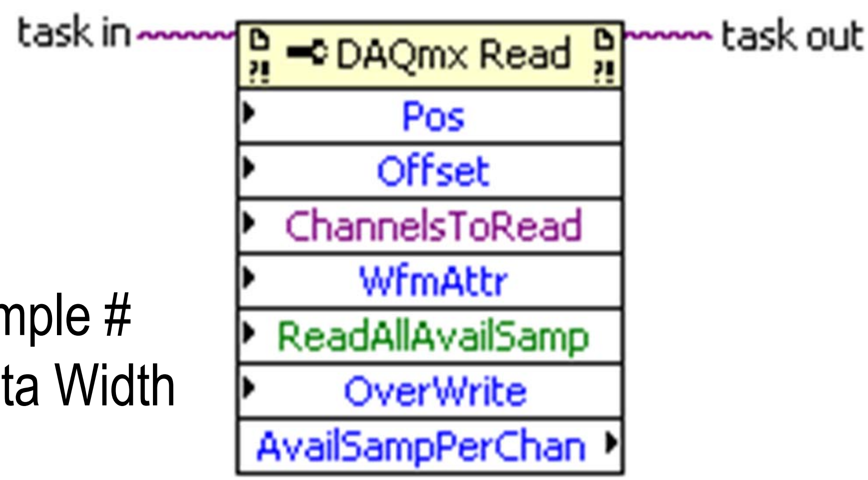
Read VI & Read Property Node

- **Read VI**
  - Used for analog, digital, counter, and raw data
  - Single or multiple channels and samples

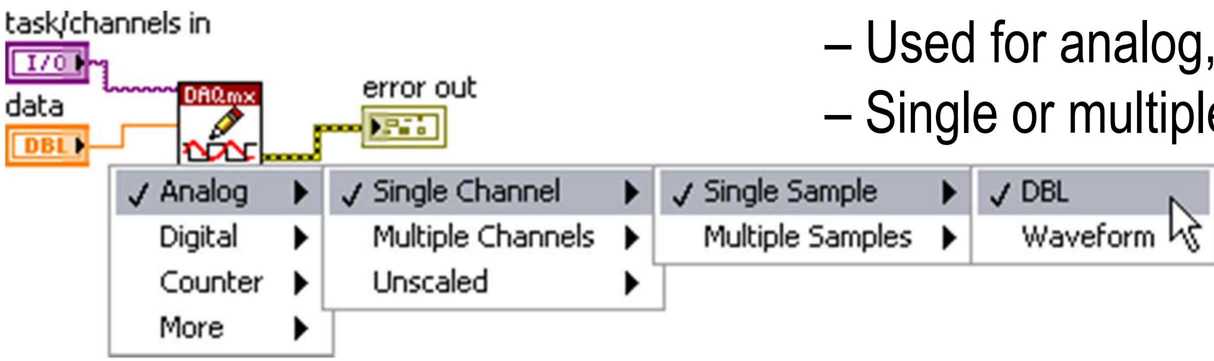


- **Properties include**
  - Offset
  - Channels to Read
  - Waveform Attributes
  - Status – Current Sample #
  - Advanced – Raw Data Width

  …And many more!

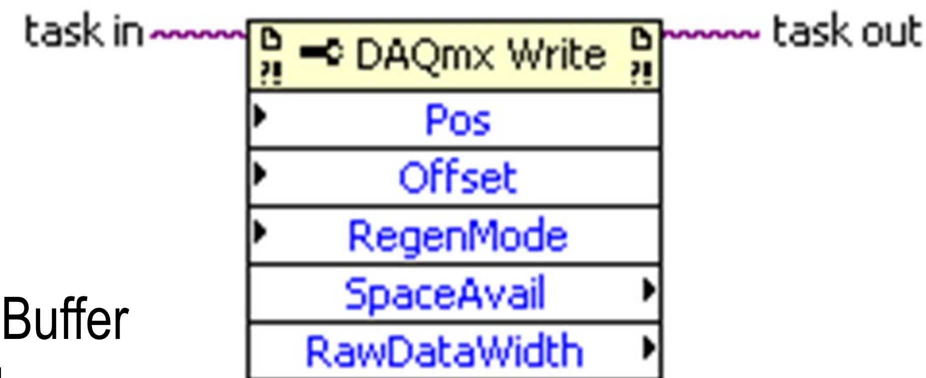


Write VI & Write Property Node



- **Write VI**
  - Used for analog, digital, and unscaled data
  - Single or multiple channels and samples

- **Properties include**
  - –Position
  - –Offset
  - –Regeneration Mode
  - –Status – Space Available in Buffer
  - –Advanced – Raw Data Width
  - …And many more!



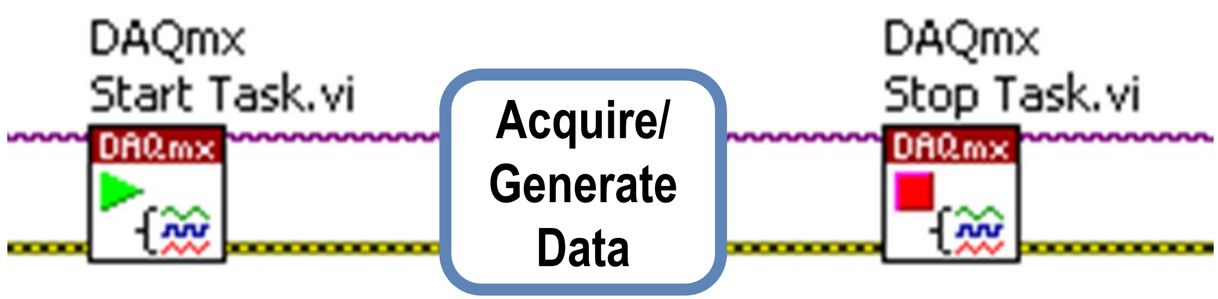
Starting and Stopping Tasks

DAQmx Start Task

- Begins measurement or generation
- Increases user control
  - Get a task ready, but do not start until desired

DAQmx Stop Task

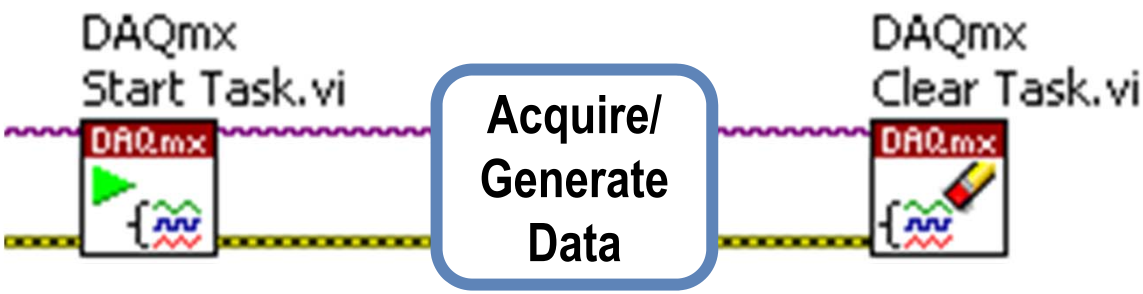
- Stops the measurement or generation
- Can restart task if stopped
- If no restart needed, then use the DAQmx Clear VI to stop and clear the task



Clearing Tasks

DAQmx Clear Task

- Stops the task if necessary
- Releases any resources the task reserved
- Use this VI to clear the task when you are finished with the task



Summary—Quiz

1. Which of the following are components of a DAQ device?
   a) Analog Input Circuitry
   b) Data Transfer Bus
   c) RAM
   d) Counter Circuitry
   e) Onboard FIFOs

Summary—Quiz Answer

1. Which of the following are components of a DAQ device?
   a) Analog Input Circuitry
   b) Data Transfer Bus
   c) RAM
   d) Counter Circuitry
   e) Onboard FIFOs

Summary—Quiz

2. All DAQmx VIs and property nodes are accessible through the NI-DAQmx palette.
   a) True
   b) False

Summary—Quiz Answer

2. All DAQmx VIs and property nodes are accessible through the NI-DAQmx palette.

   a) True

   b) False

Summary—Quiz

3. Code width defines how close your measurement is to its true value.
   a) True
   b) False

Summary—Quiz Answer

3. Code width defines how close your measurement is to its true value.
   a) True
   b) False