Data Acquisition using Measurement and Automation Explorer, DAQmx, and NI's Hardware

What is Measurement and Automation Explorer (MAX)?

NI MAX is a separate program that makes communicating with instruments as easy as possible. It is usually used before you program to LV to ensure that you can communicate with some instrument you are using.

You can communicate in MAX through USB, RS232, GPIB, etc.

Caution: I have had trouble communicating via RS-232 (use Hyperterminal or Telnet instead). However, other communication methods have been flawless.

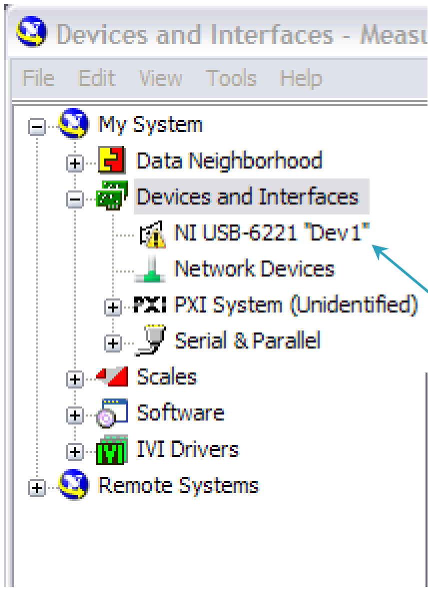
Measurement and Automation Explorer (MAX) (Ch 10, p.455)

You can also create a virtual instrument so that you can program without having the actual instrument in front of you. This way you can write programs from home without having to take the boards home.

It’s not the real thing, but it does OK.

Important: Different properties in LabVIEW populate when connecting different DAQ board types. These properties don’t show up until the DAQ board is connected. To have these properties show up without a DAQ board will require setting up a virtual instrument.

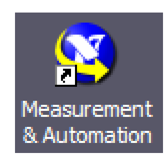
Measurement and Automation Explorer (MAX)



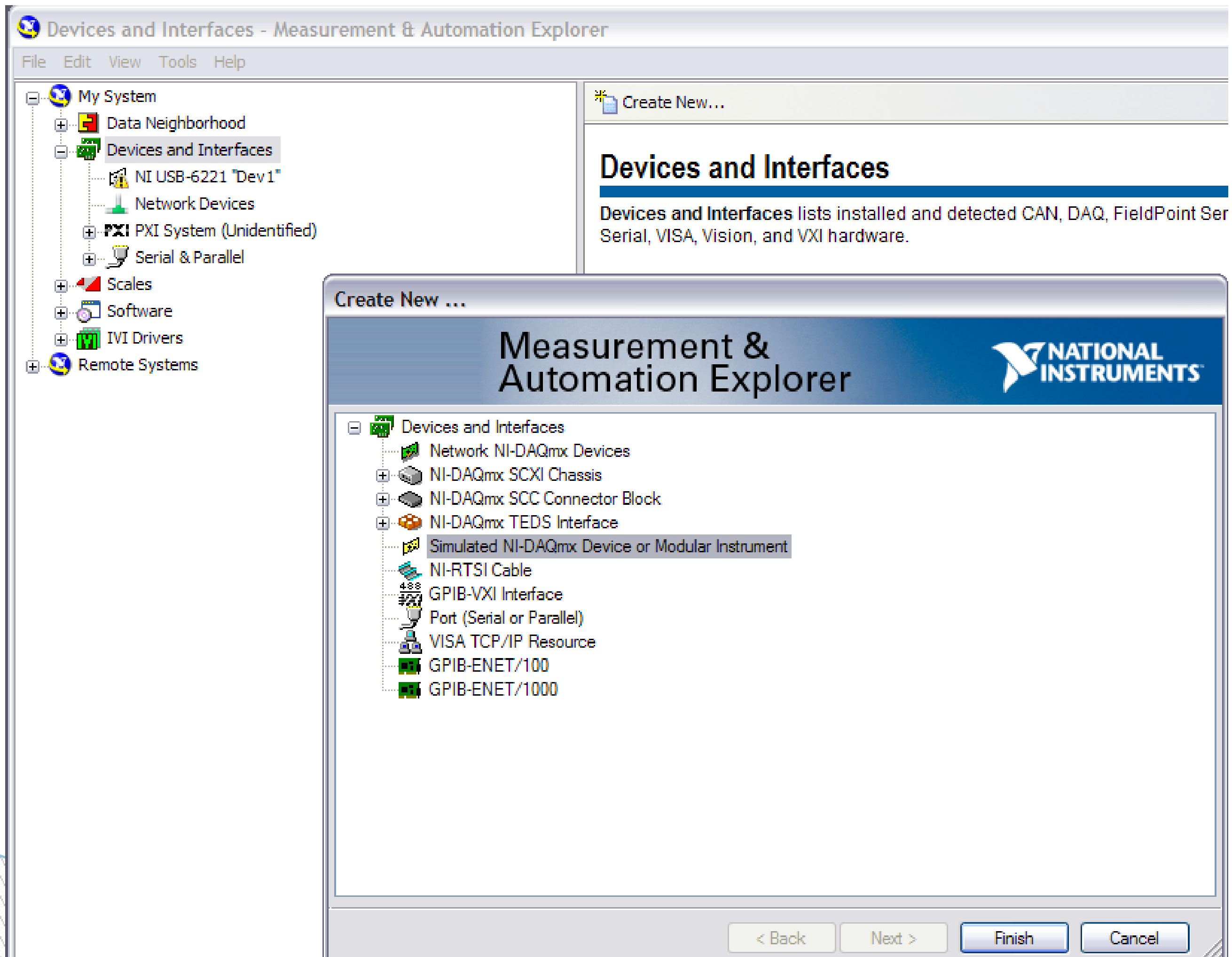
Under “Devices and Interfaces” is a list of all the devices that are available. Your instrument should be listed under this interface.

Notice that “NI USB-6221 ‘Dev 1’” has an exclamation mark beside it. This means that the Device is not connected. It will normally have a green icon beside it.

Desktop icon for MAX



MAX – Creating Virtual Instrument



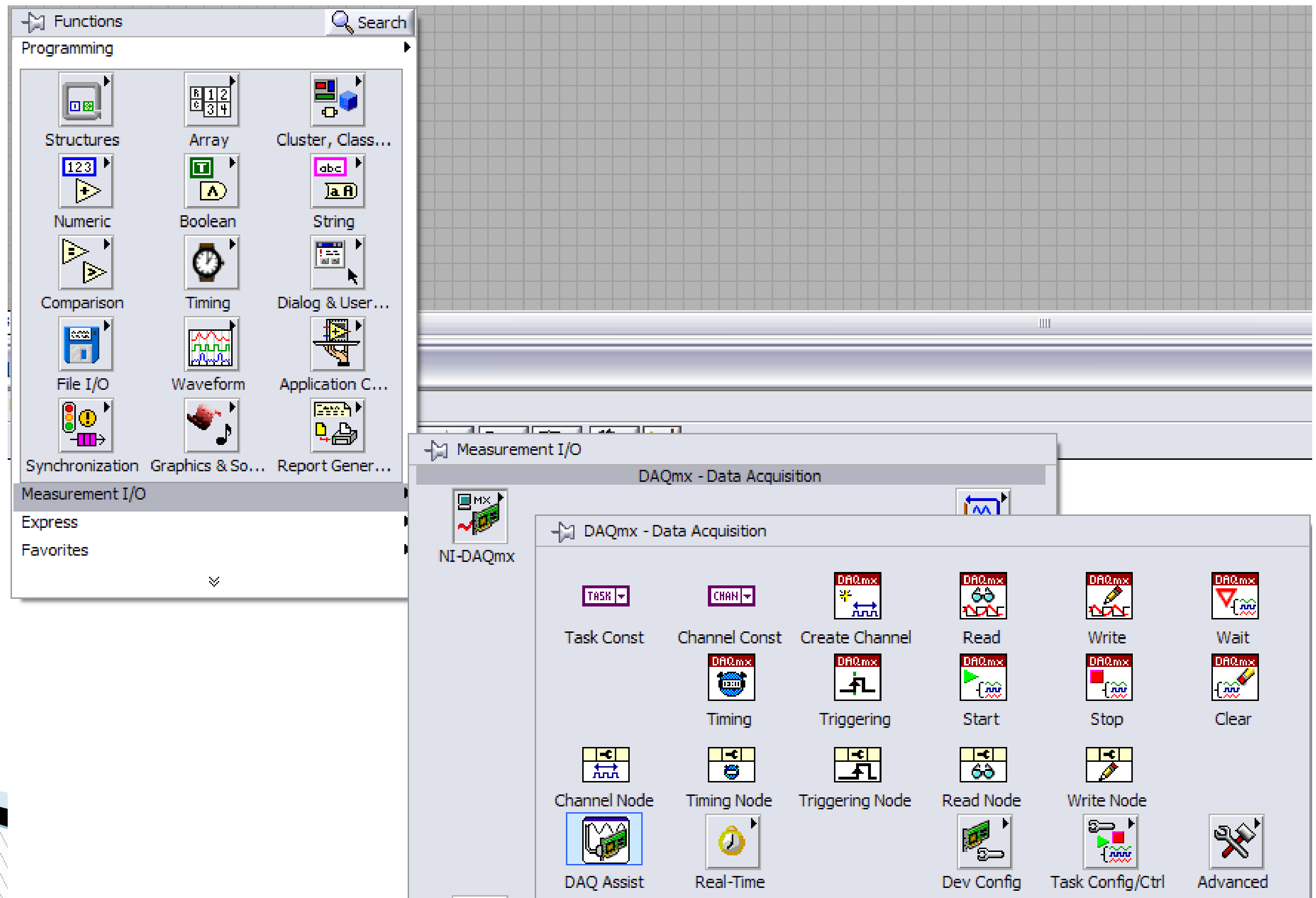
To create a virtual instrument . . .

Right-click on “Devices and Interfaces” > “Create New . . .” > “Simulated NI-DAQmx Device or Modular Instrument” > “Finish” > “M Series DAQ” > “NI USB-6221”

Now you can use a simulated version of this device (icon is yellow) without the hardware (maybe at your house or office).

DAQmx (Ch 11, p.465)

DAQmx is a cross-platform driver. You used to have download the drivers for each instrument, but now you can use the same driver for most of NI's instruments. So I can hook up a completely different board and my old code from a different instrument should work fine.



DAQmx

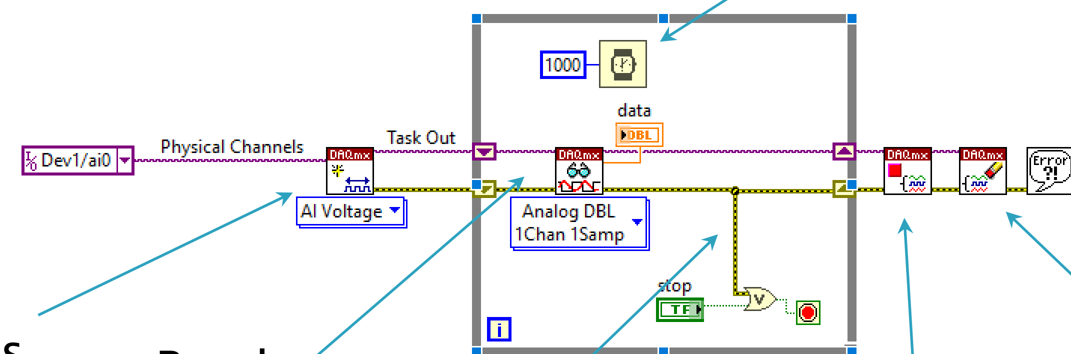
DAQ Assistant is one method for setting up your data acquisition program, but it is too "blackbox" for this course. By saying this, I mean that you lose flexibility because it can limit you from doing some advanced stuff later on in the course and so we won't use it.

DAQmx

This program reads channel 0 every second until there's an error or the user hits the stop button.

Loop every 1000 ms

Channel or Line Declaration (chooses which pins on the board to use)



Report any errors

Creates Channel

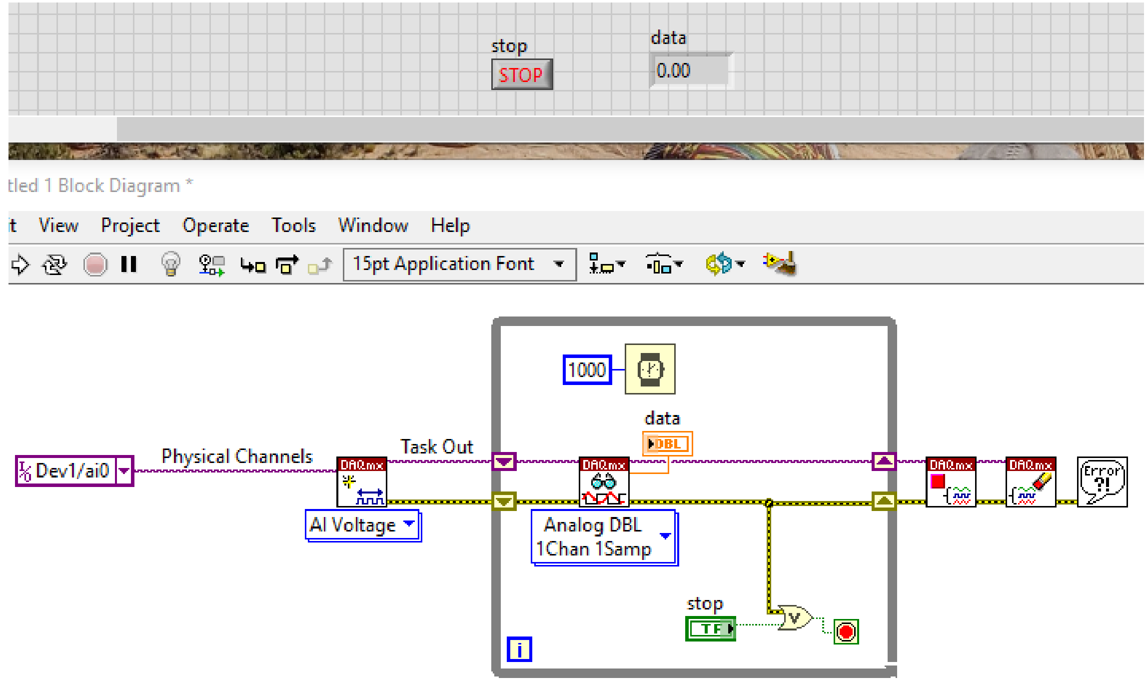
Reads voltage

Any errors stop the program *(should always have)*

Stop Reading

Dispose of (clear) channel

DAQmx



The error line serves two purposes – reports any error and makes the program sequential (flow control).

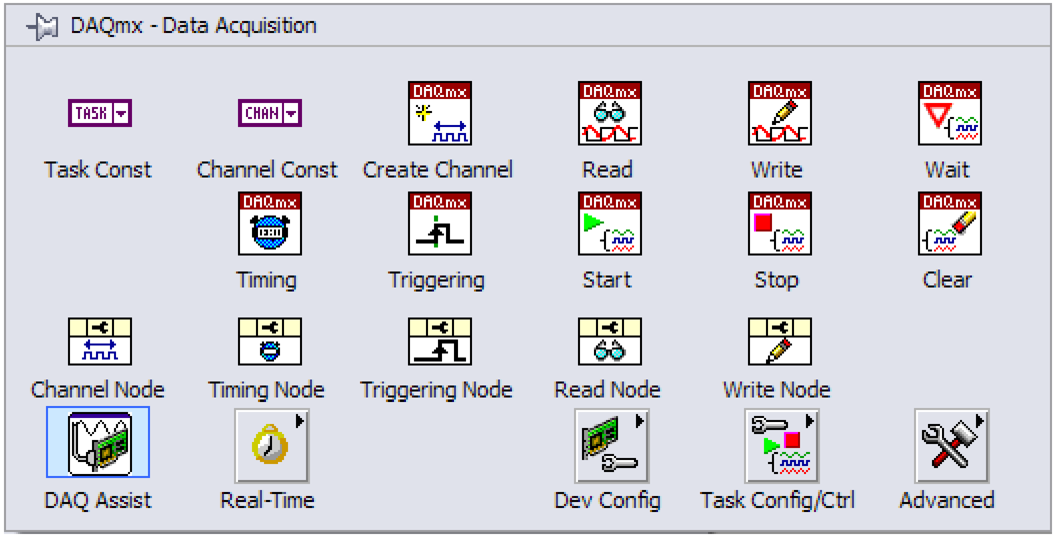
The error cluster contains status (Boolean), error code (integer), and source (string that identifies the root cause of the error).

The While Loop stops if a user hits the stop button or there exists an error. You should always have this in your code!

The read voltage is displayed in the “data” numerical indicator.

Note: This is typical code that you will use when performing data acquisition.

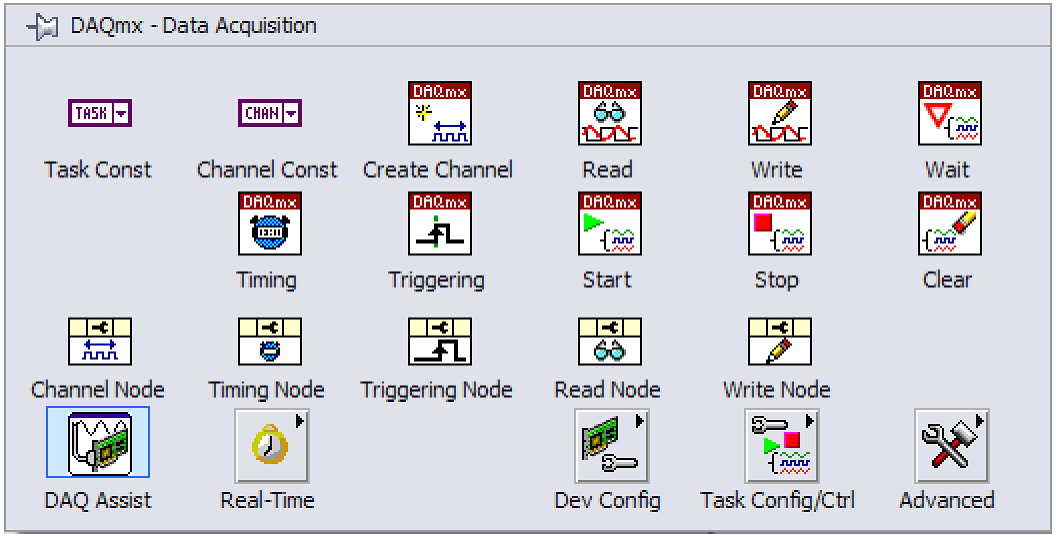
DAQmx



Write allows you to output a voltage.
Wait stops execution until a task is complete before moving on.
Timing is used to set your sampling rate and if you want to sample a finite number of times or continuously.
Start is used to begin acquisition (some VIs have an automatic start option and you don't need this, but other VIs don't).
Not shown is "**Is Task Done**" which is similar to a wait, but it doesn't stop the execution of the code. It just indicates with a Boolean if the task is done. You'll see this used a bunch in the LabVIEW examples.

With these tools mentioned, you can control an entire research lab!

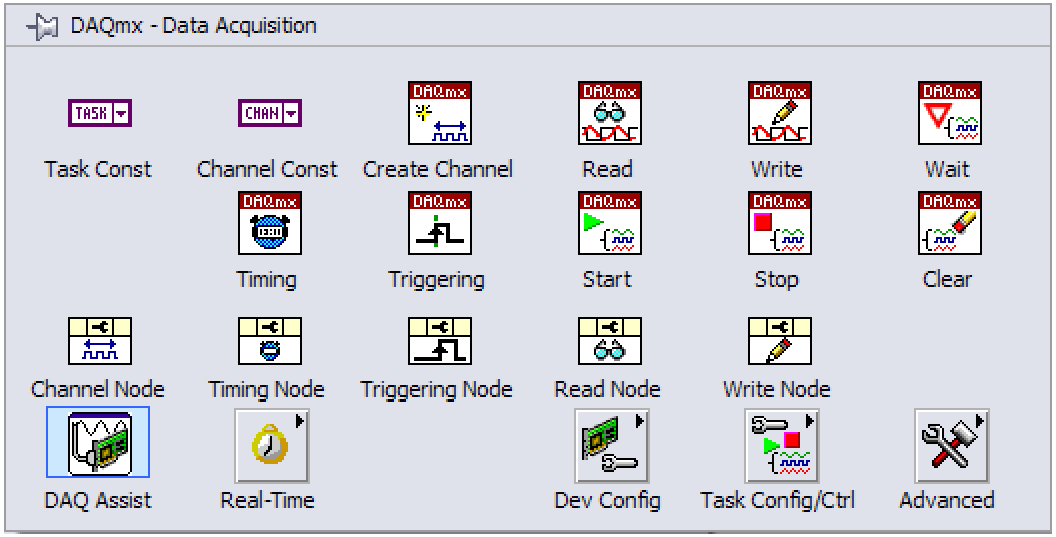
DAQmx



Note that often times you need a specific DAQ board connected to the computer **and turned on** to show which options are available for that specific DAQ board. For example, the USB-6341 board does not have a current supply associated with it where other DAQ boards would, so this option would not display when you Create Channel where other boards that support this would.

If you are working on problems outside of lab, then you could create a Simulated DAQ device (mentioned in previous slide) to have these options show up.

DAQmx



You will need to search through examples to be able to perform the specific application that you need to do. There are lots of different configurations depending on the task that needs to be performed! (Ex: you can use DAQmx for digital, analog, counters, etc.)

National Instruments Hardware

NI makes really nice hardware. Usually the hardware will last for several years once you spend the money. They have the market share for their equipment. Almost everything with National Instruments is expensive, but companies and researchers pay the price for the ease of setting up the equipment and the time that this saves.

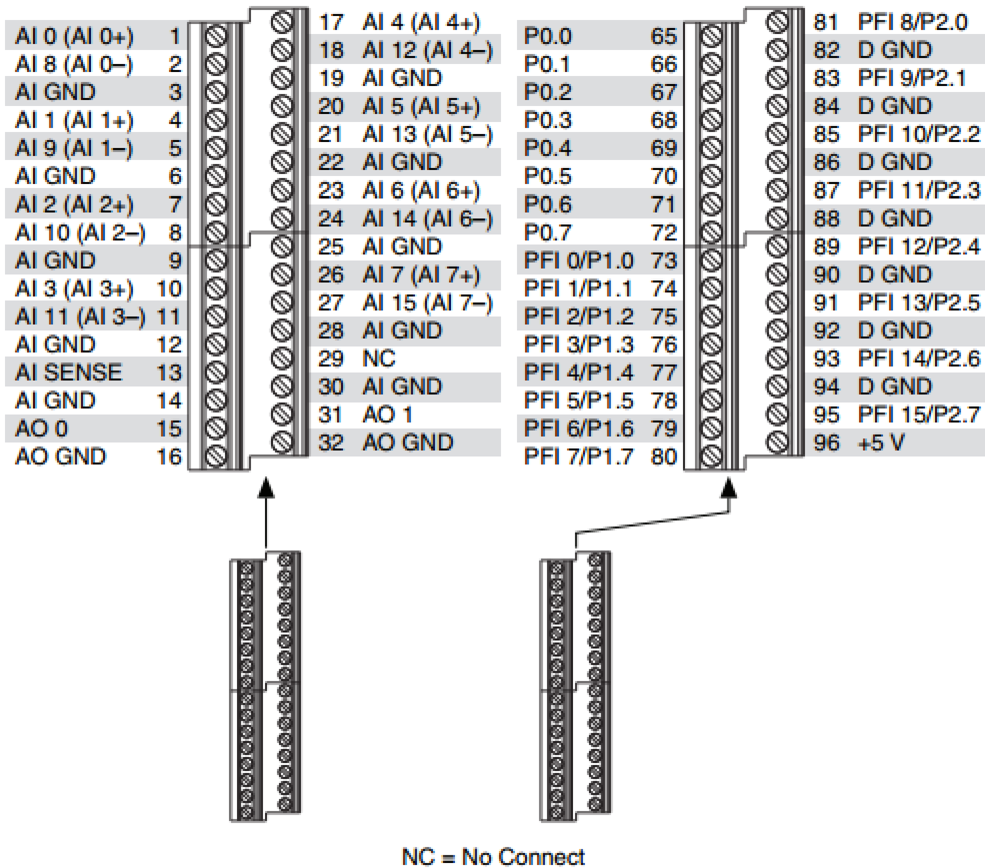
The Multifunction DAQ boards range from $150 to $3000. We're using a USB-6341 or USB-6221 Device which runs about $2800.

NI USB-6341

This board can be a power supply, function generator, variable waveform generator, counter, digital I/O, and oscilloscope in one device.

However, the device is typically slow compared to most other equipment (such as oscilloscopes, etc.).

NI USB-6341 Pinout



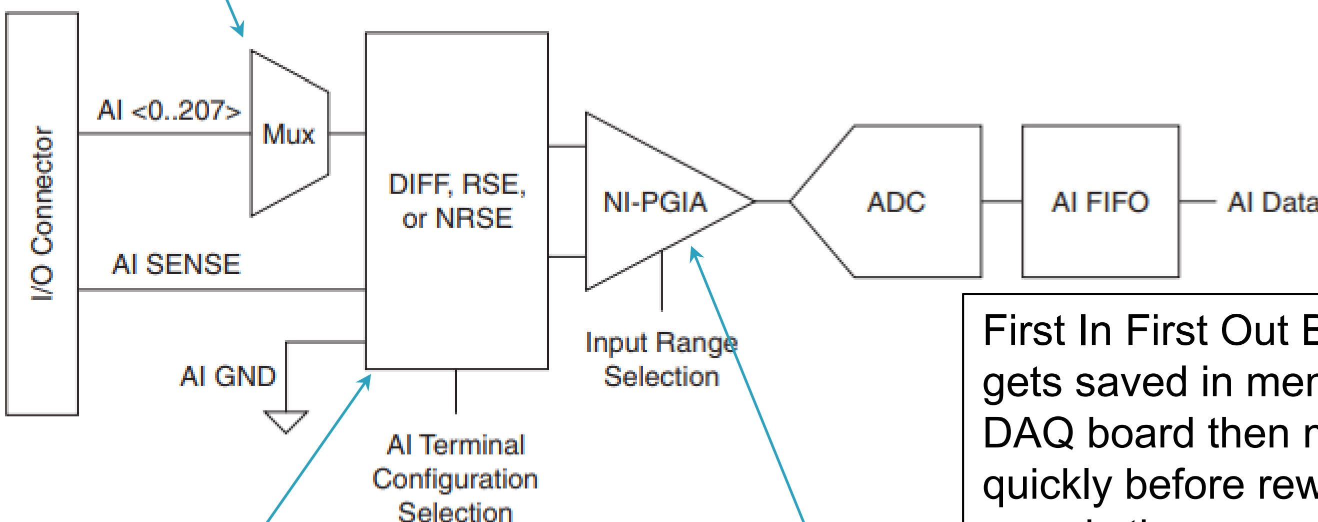
NI USB-6341 Specifications – Analog Inputs

- 8 differential or 16 single ended inputs
- 16 bit resolution
- Speeds up to 500,000 samples/sec
- 4 input ranges (±10V, ±5V, ±1V, ±0.2V)
- 10 GΩ input impedance
- FIFO buffer of 2047 samples *(Note: The DAQ board can typically do more samples than this because the buffer dumps to the computer's memory. As long as it can dump fast enough where the board doesn't overwrite values in its buffer, then you can have a larger buffer size than this.)*
- Overvoltage protection

Analog Input Port

MUX cycles between the 16 different input pins

Figure 4-1. MIO X Series Analog Input Circuitry



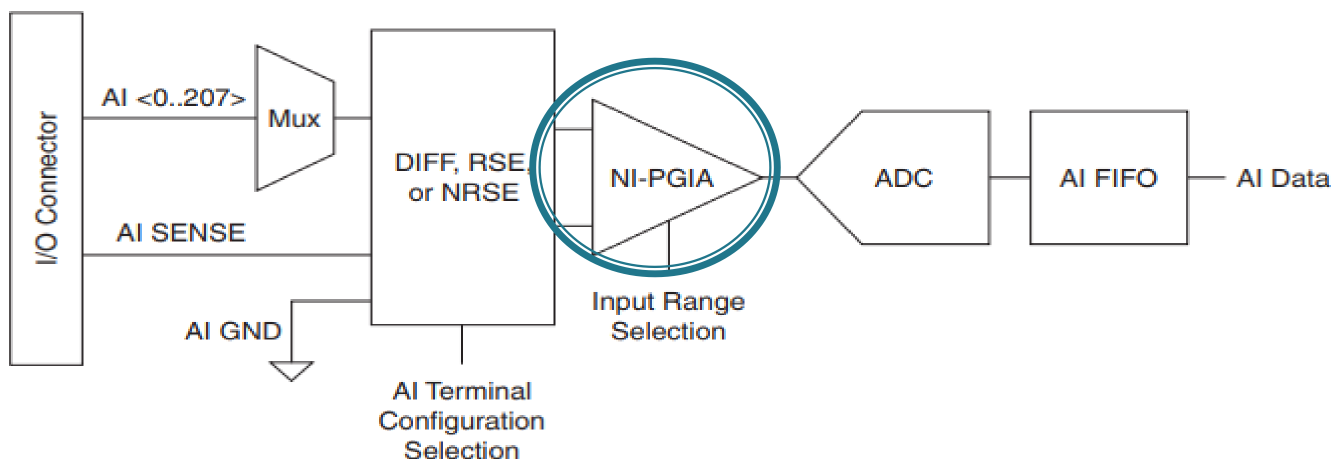
First In First Out Buffer: Data gets saved in memory on the DAQ board then must be read quickly before rewritten over or error is thrown.

Different Input Configurations: Differential, Referenced Single Ended, Non-Referenced Single Ended

Programmable Gain Instrumentation Amplifier: Changes the gain on the instrumentation amplifier to give best resolution

Analog Input Port

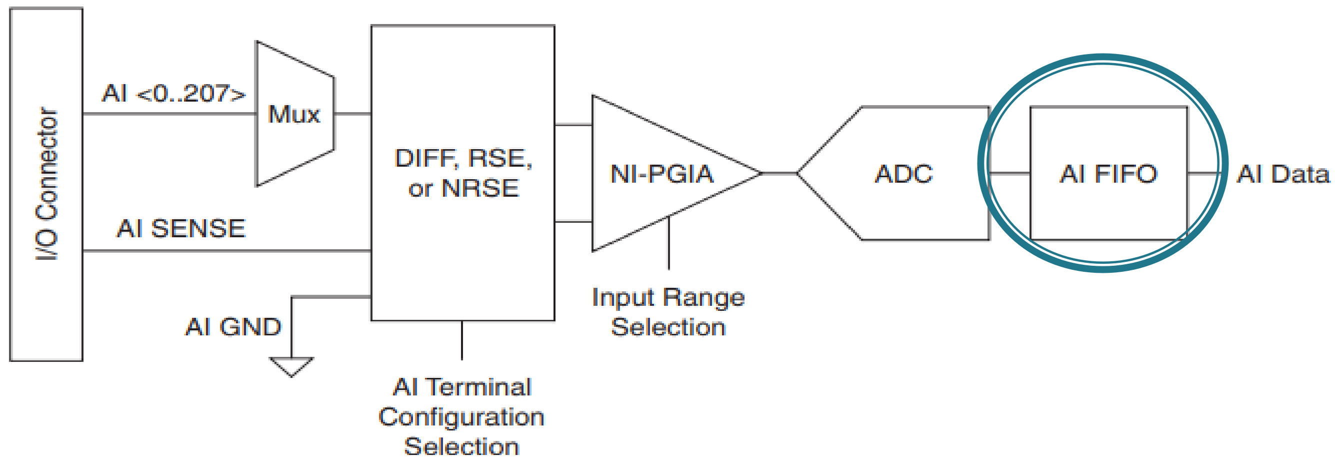
Figure 4-1. MIO X Series Analog Input Circuitry



The Programmable Gain Instrumentation Amplifier changes the gain resistor for the 4 different input ranges.

Analog Input Port

Figure 4-1. MIO X Series Analog Input Circuitry

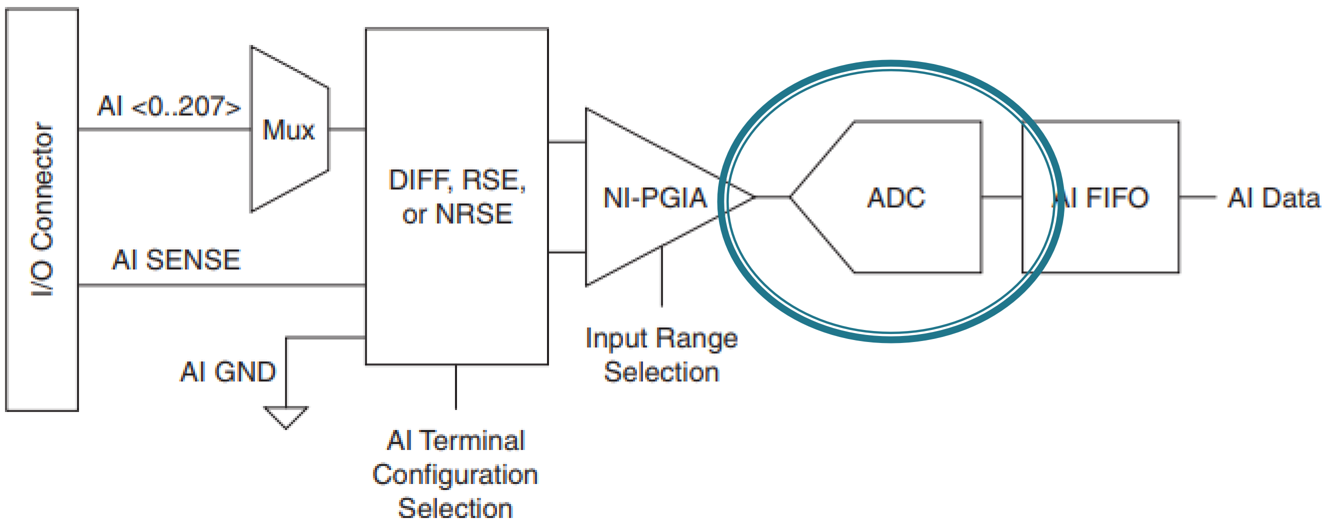


The First In First Out buffer is the on-board memory on the DAQ board. This allows the computer faster transfer rates of data because it can send a whole block of data at once instead of reading it one point at a time.

Finite Acquisition means that you read a set number of points. Continuous acquisition means that you are reading in data continuously. For continuous acquisition, you must read the buffer faster than it fills up or the DAQ board will write over previous data and an error will be thrown.

Analog Input Port

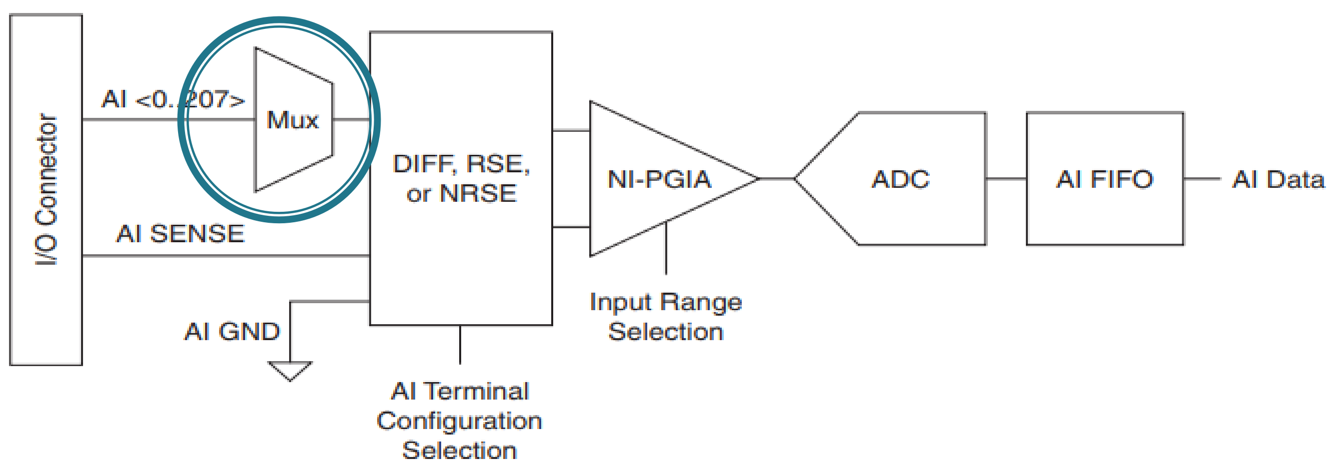
Figure 4-1. MIO X Series Analog Input Circuitry



Only one analog-to-digital converter

AI Port & Ghosting

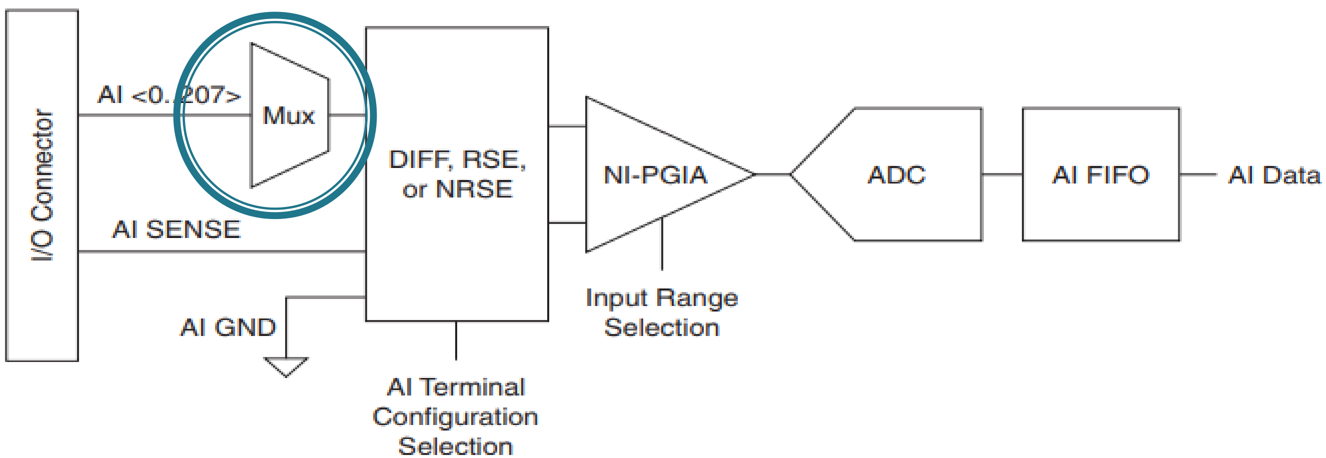
Figure 4-1. MIO X Series Analog Input Circuitry



Multiplexer cycles between the 16 different analog pins (AI0 to AI15). The way the MUX works is by switching between 16 different capacitors. When one of the channels, say channel 0, is selected, the capacitor accumulates charge. When the MUX switches to channel 1, the capacitor will leak charge backwards to channel 1. If the output impedance feeding into the pin is too large (such as the pin is not connected), then the voltage on channel 1 will be affected. This is referred to as **ghosting**. For this reason, if you are reading 4 channels and only 2 of those channels are currently hooked up, then you should ground the unused inputs or you'll see somewhat of a replica of the voltage on the previous pin.

AI Port & Ghosting

Figure 4-1. MIO X Series Analog Input Circuitry

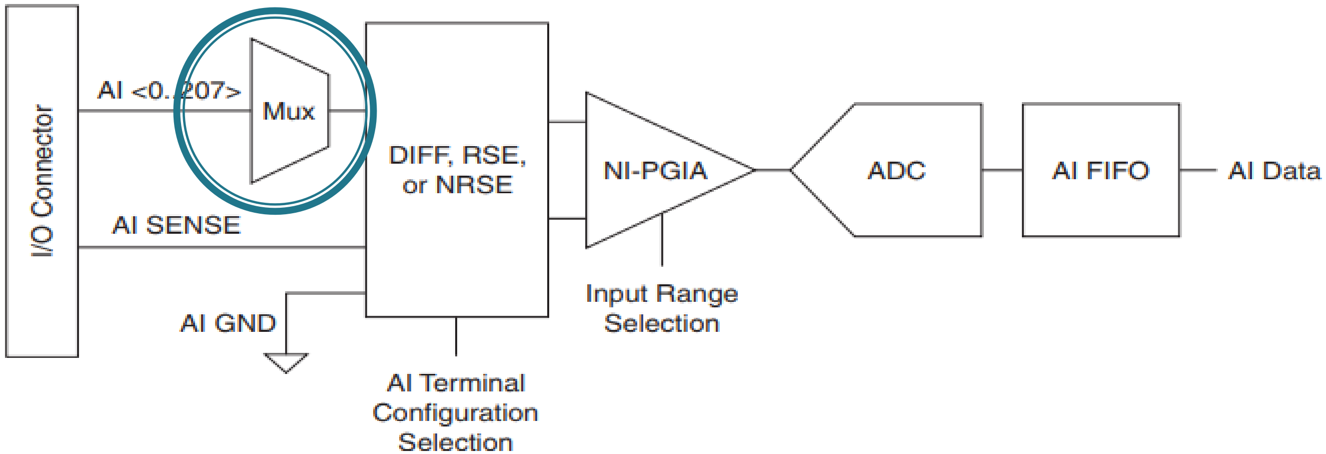


- Another way to reduce the effects of ghosting is to decrease the scan rate to allow more time for the previous voltage to settle.

- Another option is to use a voltage follower circuit to decrease the output impedance so that the charge can drain off faster.

AI Port & Ghosting

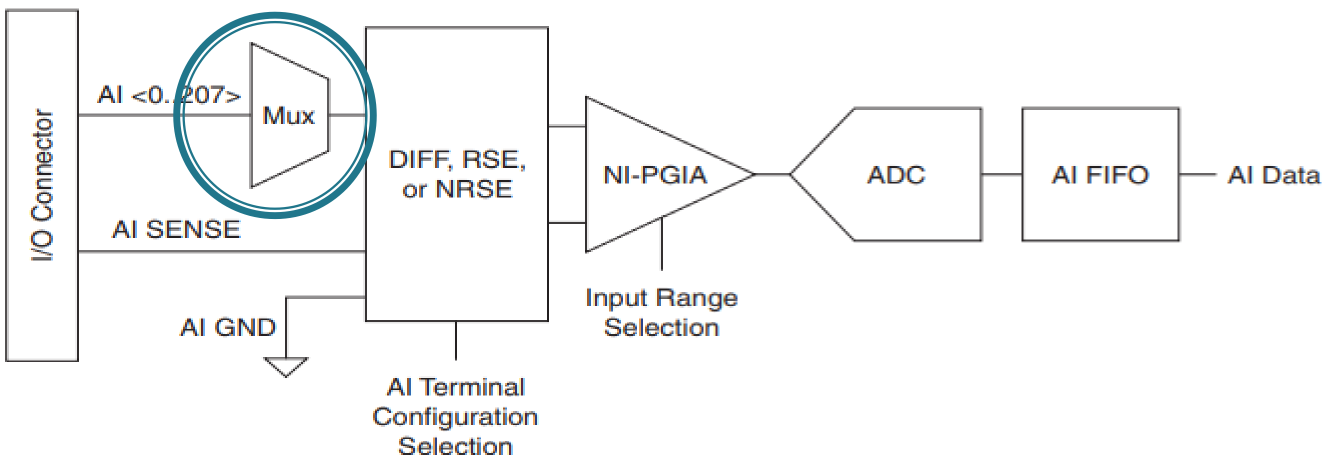
Figure 4-1. MIO X Series Analog Input Circuitry



- Another consideration is to avoid switching from a channel with a large input range to a small input range. For example, if going from 4V to 1 mV, then the new scale is 1000x smaller and can mess up the reading. *Note: reading small voltage ranges to larger voltage ranges doesn't mess up the readings as much.*
- Another option is to insert a grounded channel between signal channels.

AI Port & Ghosting

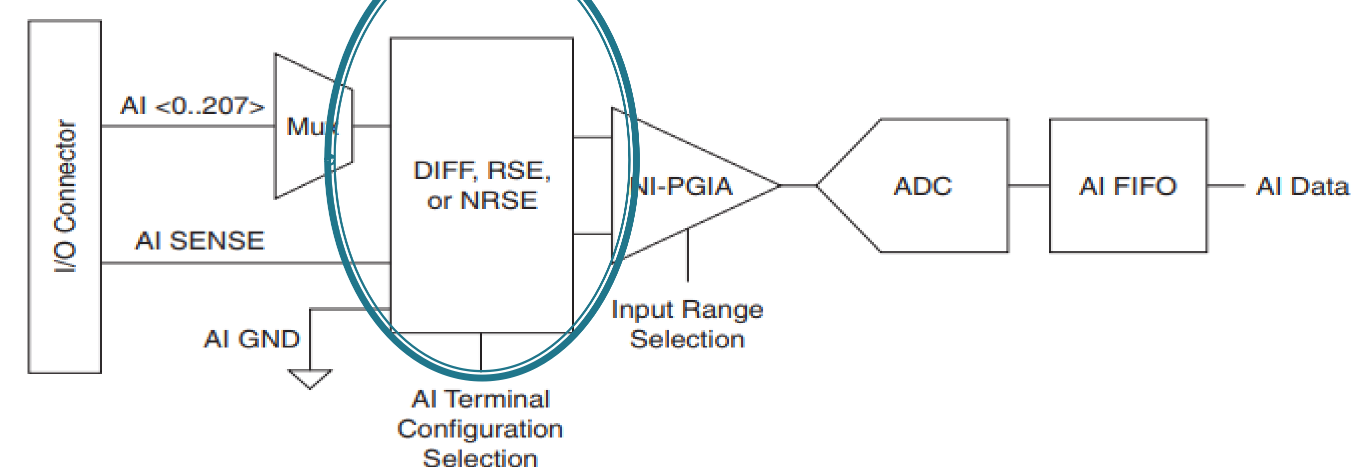
Figure 4-1. MIO X Series Analog Input Circuitry



In summary, use slower scan rates and voltage sources with smaller output impedances. Scan channels from smallest voltage ranges to largest voltage ranges. Once finished with all of the channels, provide a grounded input before repeating back to the smallest voltage range.

Analog Input Port

Figure 4-1. MIO X Series Analog Input Circuitry



There are 3 different input configurations:

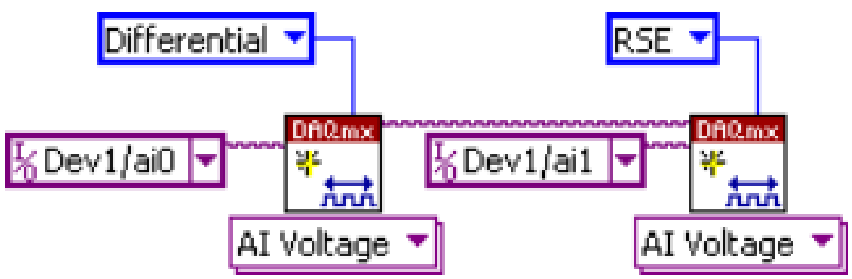
1. Differential: Measures the voltage on 2 different pins and takes the difference.
2. Reference Single Ended: Measures the voltage on a pin with ground referenced to AI Gnd.
3. Non-Referenced Single Ended: Measures the voltage on a pin with ground referenced to AI Sense.

You can program the device to have some pins as RSE and others as DIFF (or some other configuration) as well as different ranges. Flexible.

Analog Input Configuration

Differential mode has the advantage that noise picked up from the room *(60 Hz, light switch change, etc)* will be picked up on both input lines. Since we're taking the difference of the two signals, this noise is hopefully cancelled out. However, it costs you an analog input for each device you are measuring so you can only use ½ the lines (channels).

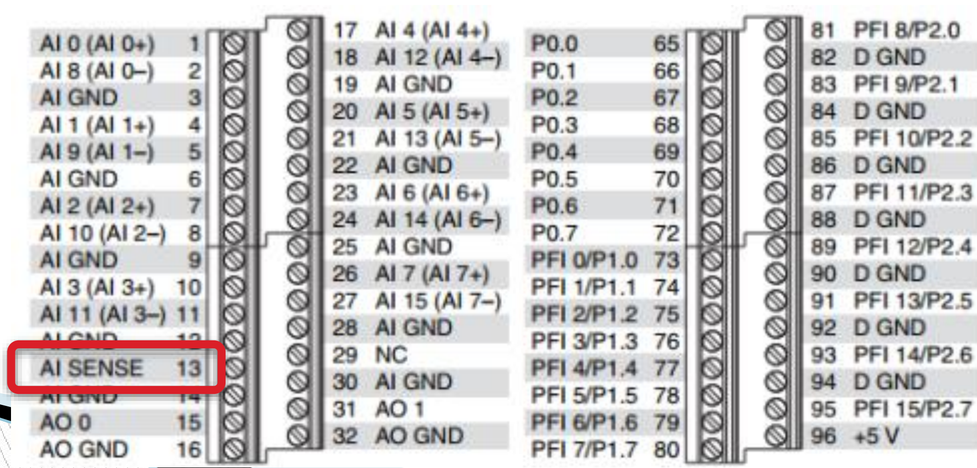
Single-Ended mode is used when all the grounds are tied together (more prone to noise, but you get twice the number of channels).



Analog Input Configuration

Referenced Single-Ended mode is when all the instruments are tied to the same ground including the DAQ board's ground (AI Ground). This mode is not recommended unless we are looking at a floating voltage (the power supply is floating b/c of the transformer, but the function generator and oscilloscope are not (it's tied to earth ground)).

Non-Referenced Single-Ended mode is when again all the instruments are tied to the same ground, but instead of using the DAQ board's AI ground, you tie the instruments to the pin AI Sense.



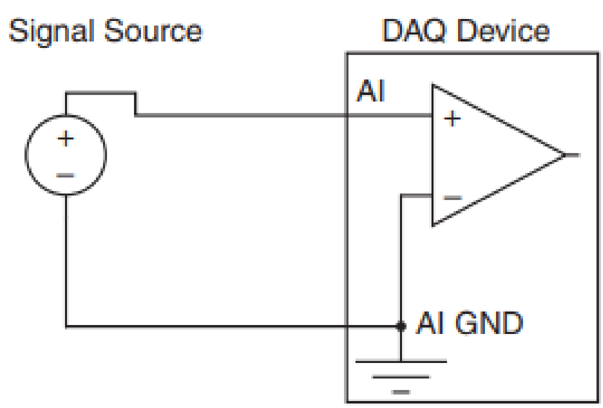
See next slides for diagrams.

AI Configuration Using Floating Input Signals >1V

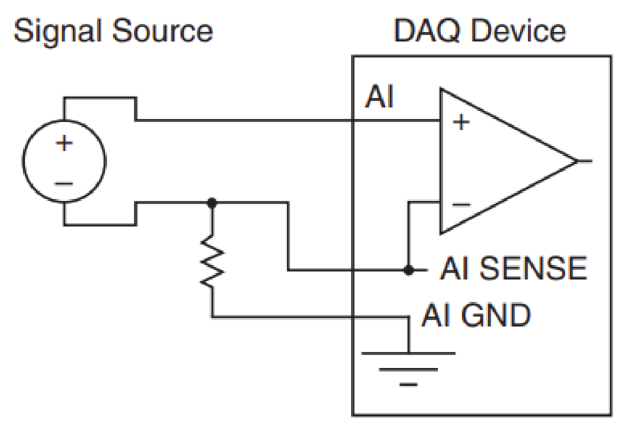
An example of a floating input signal would be a battery, thermocouple, transformer (could be from a power supply), or anything that's isolated from the building's ground.

When reading a floating voltage, it is best to use single-ended mode if the signal is >1V (such as a battery or certain sensors), in a **non-noisy environment**, and all the grounds of different signal sources can be connected together.

RSE



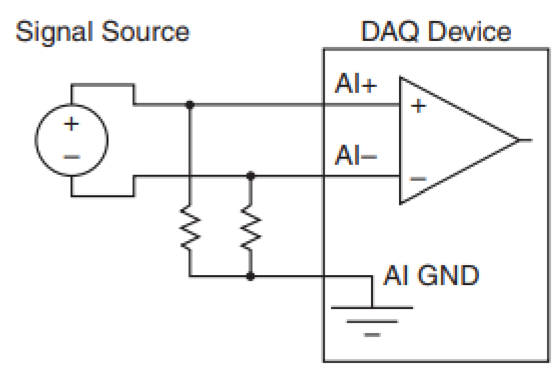
NRSE



AI Configuration Using Floating Input Signals <1V

Use Differential Mode when dealing with floating voltages that are small (<1V), in a **noisy environment**, or using long leads (>3m). An example would be reading a thermocouple.

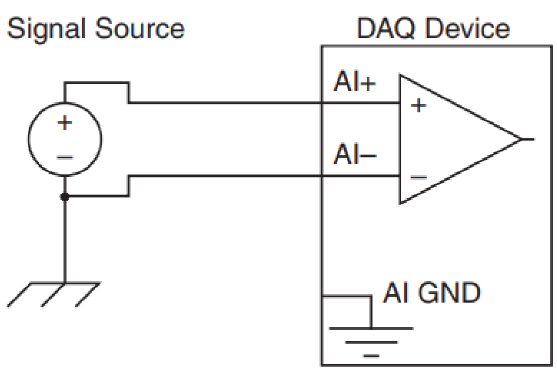
The source voltage may float up to a level greater than the DAQ board's input range, so you need to add in resistors to reference this source. The resistor should be about 100x greater than the output impedance of the source (typically in the 1k to 100k range).



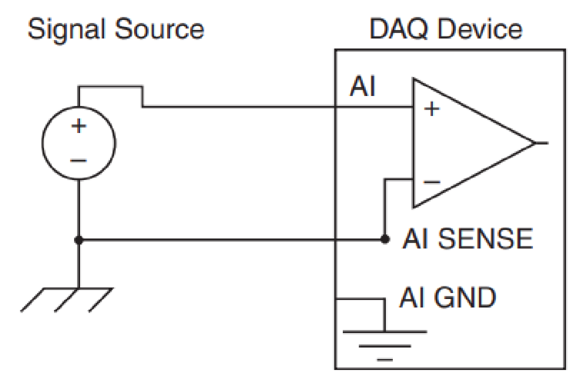
AI Configuration Grounded-Referenced Signals

If you are dealing with signals that are tied to earth ground (such as our function generators), then use either Differential or Non-Referenced Single-Ended. Notice that RSE is not recommended because of the potential for ground loop noise.

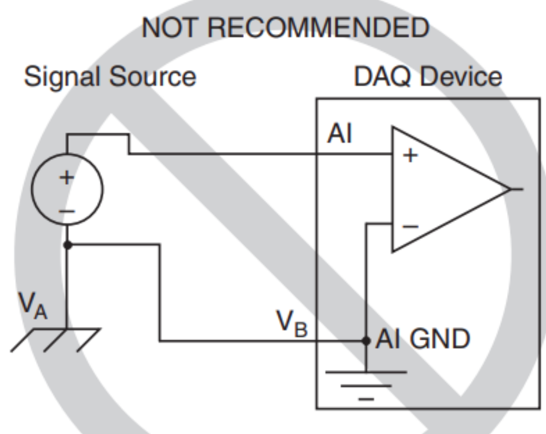
Differential



NRSE



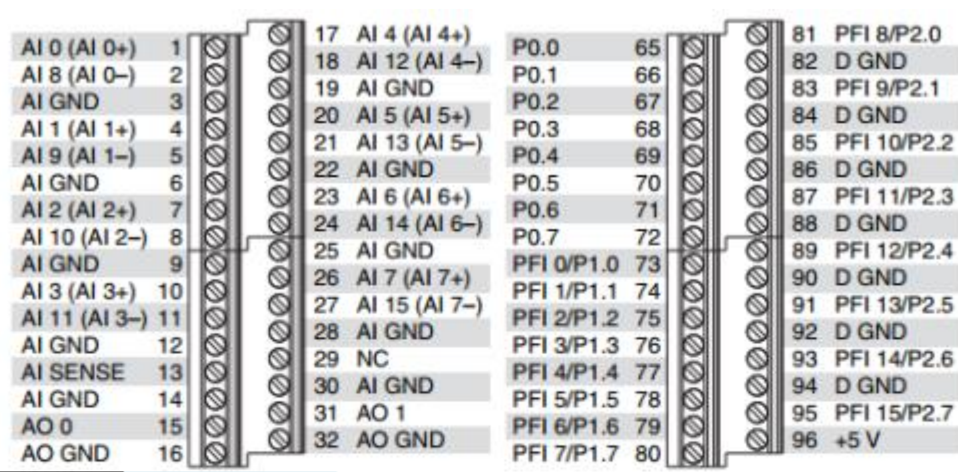
RSE



Ground-loop potential ($V_A - V_B$) are added to measured signal.

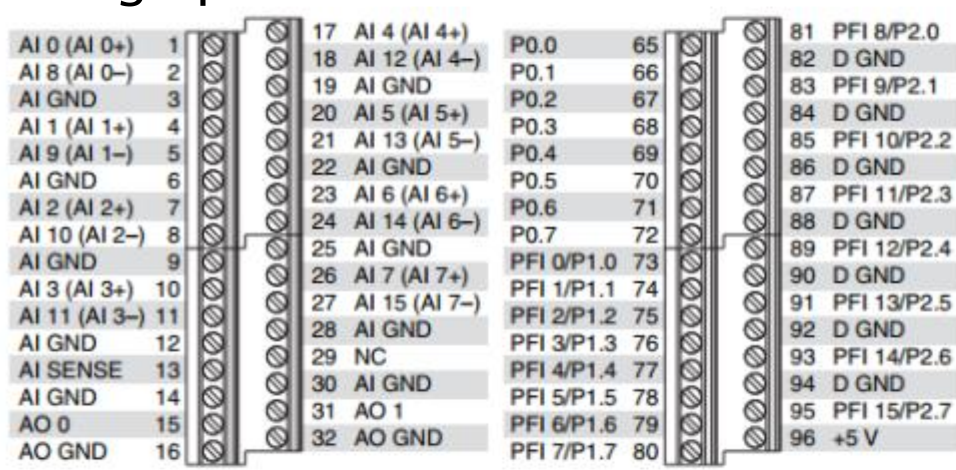
NI USB-6341 Specifications – Analog Outputs

- 2 Analog Outputs
- 16 bit resolution
- 900 kS/s for 1 chan, 840 kS/s for 2 chan
- Output ±10V
- Output impedance of 0.2 Ω
- 5 mA output
- Buffer size of 8191 samples



NI USB-6341 Specifications – Digital I/O/PFI

- 3 ports (port 0 - 2). All 3 ports contain 8 bits.
- Port 0 contains 8 Digital I/O
- Port 1 & 2 both contain either 8 Digital I/O or 8 PFI per port
- PFI stands for Programmable Function Interface. PFI can be used as either an input or output timing signal for Analog/Digital/Counter/Timer functions (clock or trigger).
- ±20V input voltage protection



NI USB-6341 Specifications – General Purpose Counters/Timers

- 4 counters/timers
- 32 bit resolution (means it will count to 4G)
- Counter: Edge counting, pulse width, period, two-edge separation
- Timer: Pulses, Pulse train, Frequency Divider
- 3 Base Clocks: 100 MHz, 20 MHz, 100 kHz

- The maximum frequency of the counter for the 6341 board is 25 MHz!

All of the Counters/Timers pins are on the Digital side of the DAQ board.

NI USB-6341 Specifications – NI-DAQ Counter/Timer Pins

Source: An input that can change the current count of the counter's register. Looks for an edge. Settings such as rising or falling edge and count up or down is programmed in software.

Gate: An input that decides if an edge on the source input will change the count.

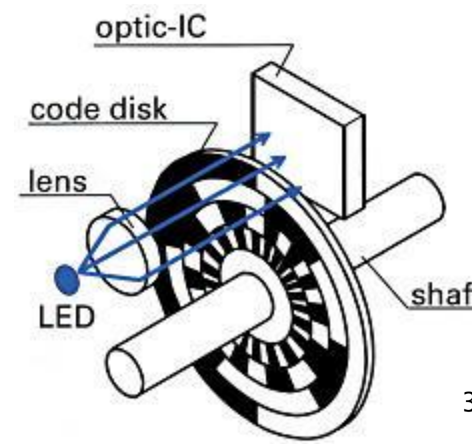
Out: An output to the counter. Used when creating pulses or pulse trains, etc.

Aux: First input terminal when measuring Two-Edge Separation. Can also be used to determine if the count should count up or down.

A: Used for quadrature encoder position measurements.

B: Also used for quadrature encoder position measurements. Phase of A and B determines if the wheel is moving clockwise or counter-clockwise which determines if the count should count up or down.

Z: Used for quadrature encoder to zero the position measurement.



NI USB-6341 Specs – Default NI-DAQ Counter/Timer **Default** Pins - not recommended

These are the default pins, but **a better approach is to define the Programmable Function Interface (PFI) pin in code**. This way if the user switches to a different type of DAQ board, then it makes wiring easier because they don't have to look up default pins.

Counter/Timer Signal	Default Terminal Name
CTR 0 SRC	PFI 8
CTR 0 GATE	PFI 9
CTR 0 AUX	PFI 10
CTR 0 OUT	PFI 12
CTR 0 A	PFI 8
CTR 0 Z	PFI 9
CTR 0 B	PFI 10
CTR 1 SRC	PFI 3
CTR 1 GATE	PFI 4
CTR 1 AUX	PFI 11
CTR 1 OUT	PFI 13
CTR 1 A	PFI 3
CTR 1 Z	PFI 4
CTR 1 B	PFI 11
CTR 2 SRC	PFI 0
CTR 2 GATE	PFI 1
CTR 2 AUX	PFI 2
CTR 2 OUT	PFI 14
CTR 2 A	PFI 0
CTR 2 Z	PFI 1
CTR 2 B	PFI 2
CTR 3 SRC	PFI 5
CTR 3 GATE	PFI 6
CTR 3 AUX	PFI 7
CTR 3 OUT	PFI 15
CTR 3 A	PFI 5
CTR 3 Z	PFI 6
CTR 3 B	PFI 7
FREQ OUT	PFI 14

NI USB-6341 Specifications – Default Counters Connections

You can use LabVIEW's help "Connecting Counter Signals" to figure out the pins to use to make counter connections or a **better approach is to allow the user control to set the pins**. Then if you switch DAQ boards, the pin #'s don't matter since the user sets these.

X Series Signal Connections for Counters

The following table lists the default input terminals for various counter measurements on X Series devices. You can use a different PFI line for any of the input terminals. To change the PFI input for a measurement, use the NI-DAQmx channel attributes/properties.

Measurement	Ctr0	Ctr1	Ctr2	Ctr3
Count Edges	Edges: PFI 8 Count Direction: PFI 10	Edges: PFI 3 Count Direction: PFI 11	Edges: PFI 0 Count Direction: PFI 2	Edges: PFI 5 Count Direction: PFI 7
Pulse Width Measurement	PFI 9	PFI 4	PFI 1	PFI 6
Period/Frequency Measurement (Low Frequency with One Counter)	PFI 9	PFI 4	PFI 1	PFI 6
Period/Frequency Measurement (High Frequency with Two Counters)	PFI 8	PFI 3	PFI 0	PFI 5
Period/Frequency Measurement (Large Range with Two Counters)	PFI 8	PFI 3	PFI 0	PFI 5
Pulse Measurement	PFI 9	PFI 4	PFI 1	PFI 6
Semiperiod Measurement	PFI 9	PFI 4	PFI 1	PFI 6
Two-Edge Separation Measurement	Start: PFI 10 Stop: PFI 9	Start: PFI 11 Stop: PFI 4	Start: PFI 2 Stop: PFI 1	Start: PFI 7 Stop: PFI 6
Position Measurement	A: PFI 8 B: PFI 10 Z: PFI 9	A: PFI 3 B: PFI 11 Z: PFI 4	A: PFI 0 B: PFI 2 Z: PFI 1	A: PFI 5 B: PFI 7 Z: PFI 6

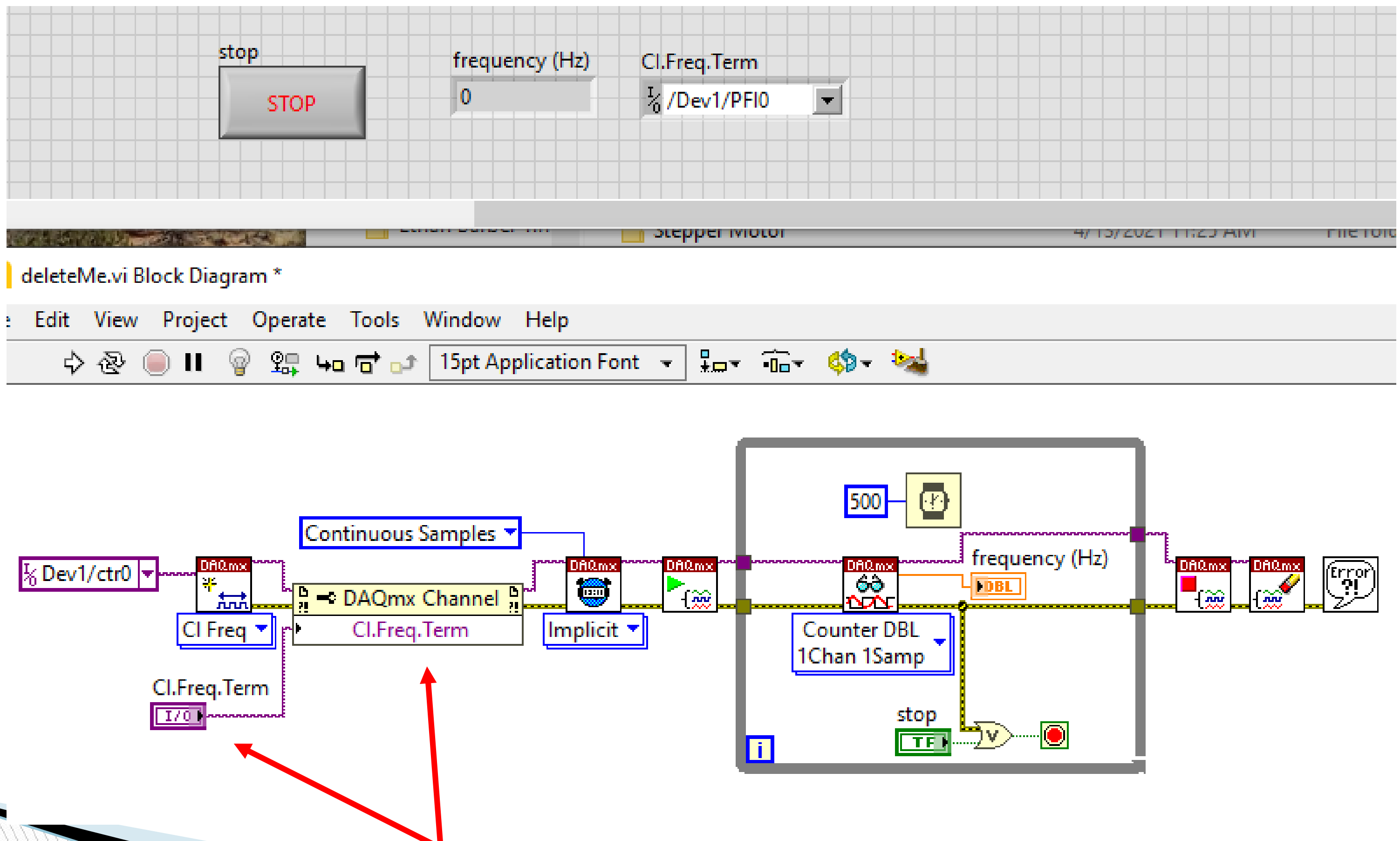
The following table lists the output terminals for counter output. You can use a different PFI line for any of the output terminals.

Ctr0	Ctr1	Ctr2	Ctr3
PFI 12	PFI 13	PFI 14	PFI 15

Parent topic: Connecting Counter Signals

NI USB-6341 Specifications – Manually Changing Counters Connections

This code measures frequency on the user defined PFI0 pin (instead of using the default pin).



NI USB-6341 Specifications – Frequency Generator

- 1 Frequency Generator
- Base clocks of 20 MHz, 10 MHz, 100 kHz
- Divisors: 1 to 16

NI USB-6221

This board is the older version for the USB-6341.

NI USB-6221 Pinout

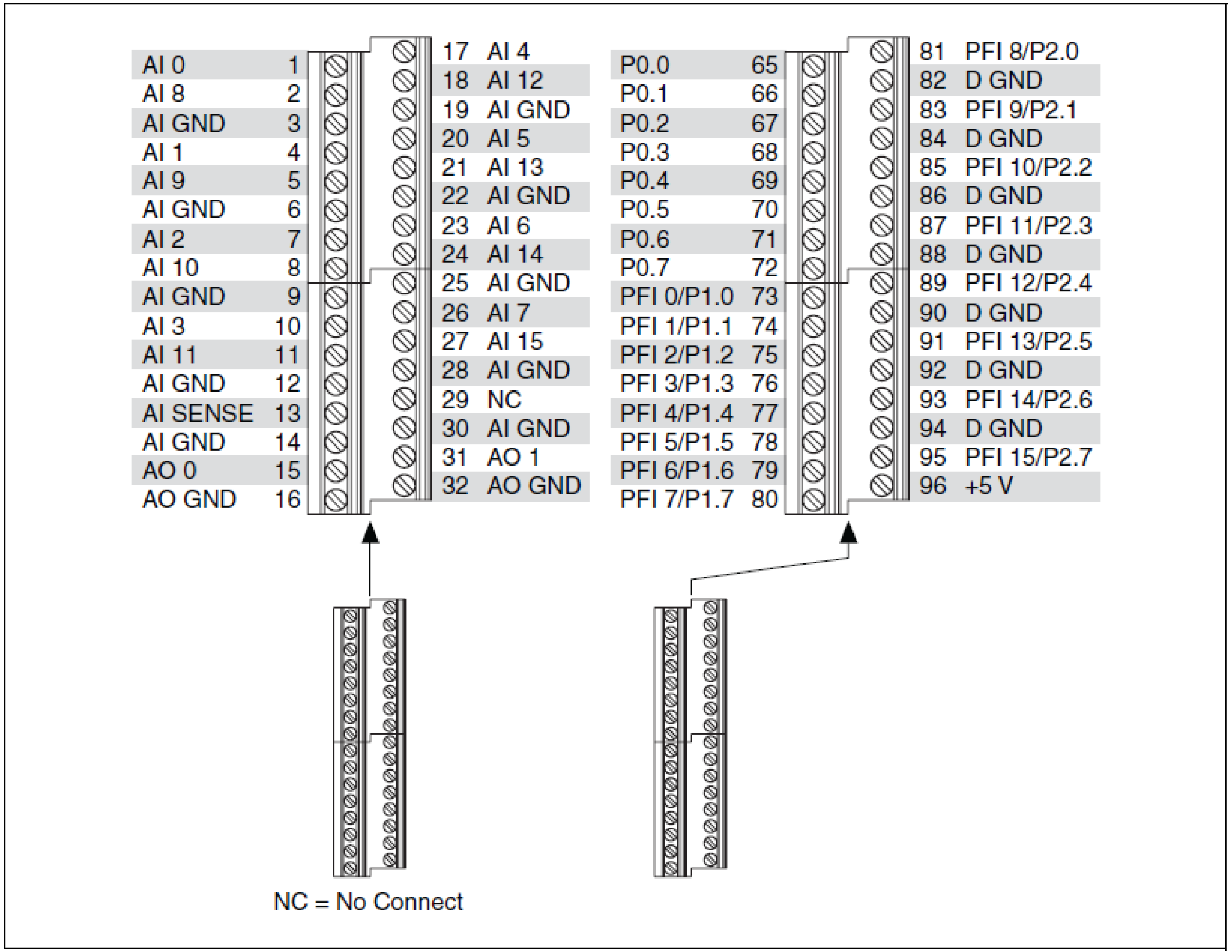


Figure 4. USB-6221 Screw Terminal Pinout

NI USB-6221 Specifications – Analog Inputs

- 8 differential or 16 single ended inputs
- 16 bit resolution
- 250,000 samples/sec
- 4 input ranges (±10V, ±5V, ±1V, ±0.2V)
- 10 GΩ input impedance
- Buffer of 4095 samples
- Overvoltage protection

NI USB-6221 Specifications – Analog Outputs

- 2 Analog Outputs
- 16 bit resolution
- 833 kS/s for 1 chan, 740 kS/s for 2 chan
- Output ±10V
- Output impedance of 0.2 Ω
- 5 mA output
- Buffer size of 8191 samples

NI USB-6221 Specifications – Digital I/O/PFI

- 2 ports (port 0 and port 1)
- Port 0 contains 8 Digital I/O
- Port 1 contains either 8 Digital I/O or 8 PFI
- PFI stands for Programmable Function Interface. PFI can be used as either an input or output timing signal for Analog/Digital/Counter/Timer functions (clock or trigger).
- ±20V input voltage protection

NI USB-6221 Specifications – General Purpose Counters/Timers

- 2 counters/timers
- 32 bit resolution
- Counter: Edge counting, pulse width, period, two-edge separation
- Timer: Pulses, Pulse train, Frequency Divider
- 3 Base Clocks: 80 MHz, 20 MHz, 100 kHz

NI USB-6221 Specifications – Default NI-DAQmx Counter/Timer Pins

Source: An input that can change the current count of the counter's register. Looks for an edge. Settings such as rising or falling edge and count up or down is determined in software.

Gate: An input that decides if an edge on the source input will change the count.

Out: An output to the counter. Used when creating pulses or pulse trains.

Aux: First input terminal when measuring Two-Edge Separation. Can also be used to determine if the count should count up or down.

B: Determines if the count should count up or down. Also used for quadrature encoder position measurements.

A: Used for quadrature encoder position measurements.

Z: Used for quadrature encoder to zero the position measurement.

NI USB-6221 Specifications – Default NI-DAQmx Counter/Timer Pins

Table A-2. Default NI-DAQmx Counter/Timer Pins

Counter/Timer Signal	Default Pin Number (Name)
CTR 0 SRC	37 (PFI 8)
CTR 0 GATE	3 (PFI 9)
CTR 0 AUX	45 (PFI 10)
CTR 0 OUT	2 (PFI 12)
CTR 0 A	37 (PFI 8)
CTR 0 Z	3 (PFI 9)
CTR 0 B	45 (PFI 10)
CTR 1 SRC	42 (PFI 3)
CTR 1 GATE	41 (PFI 4)
CTR 1 AUX	46 (PFI 11)
CTR 1 OUT	40 (PFI 13)
CTR 1 A	42 (PFI 3)
CTR 1 Z	41 (PFI 4)
CTR 1 B	46 (PFI 11)
FREQ OUT	1 (PFI 14)

NI USB-6221 Specifications – Counters Connections

You can use LabVIEW's help "Connecting Counter Signals" to figure out the pins to use to make counter connections.

68-Pin M Series Signal Connections for Counters

The following table lists the default input terminals for various counter measurements on M Series devices, including M Series USB devices, such as the NI USB 6259 screw terminal and NI USB-6229 BNC devices. You can use a different PFI line for any of the input terminals. To change the PFI input for a measurement, use the NI-DAQmx channel attributes/properties.

Measurement	Ctr0	Ctr1
Count Edges	Edges: PFI 8 Count Direction: PFI 10	Edges: PFI 3 Count Direction: PFI 11
Pulse Width Measurement	PFI 9	PFI 4
Period/Frequency Measurement (Low Frequency with One Counter)	PFI 9	PFI 4
Period/Frequency Measurement (High Frequency with Two Counters)	PFI 8	PFI 3
Period/Frequency Measurement (Large Range with Two Counters)	PFI 8	PFI 3
Semiperiod Measurement	PFI 9	PFI 4
Two-Edge Separation Measurement	Start: PFI 10 Stop: PFI 9	Start: PFI 11 Stop: PFI 4
Position Measurement	A: PFI 8 B: PFI 10 Z: PFI 9	A: PFI 3 B: PFI 11 Z: PFI 4

The following table lists the output terminals for counter output. You can use a different PFI line for any of the output terminals.

Ctr0	Ctr1
PFI 12	PFI 13

NI USB-6221 Specifications – Frequency Generator

- 1 Frequency Generator
- Base clocks of 10 MHz, 100 kHz