NI-DAQmx Device Considerations

January 2008, 370738M-01

This help file contains information specific to analog output (AO) Series devices, C Series, B Series, E Series devices, digital I/O (DIO) devices, DSA devices, M Series devices, S Series devices, SCC devices, SCXI devices, switches, timing I/O (TIO) devices, and USB DAQ devices that might help you as you create applications with NI-DAQmx.

This document describes only NI-DAQmx. For information on Traditional NI-DAQ (Legacy), refer to the *Traditional NI-DAQ (Legacy) User Manual* or the *Traditional NI-DAQ (Legacy) Readme*.

Analog Triggering

This section contains information about analog triggering for C Series, DSA, E Series, M Series, and S Series devices.

Valid Analog Trigger Sources for DSA Devices

The analog trigger source must be a channel included in your physical channel list. PFI 0 is not a valid analog trigger source. PFI 0 is reserved for digital triggers.

Analog Triggering Considerations for C Series, E Series, M Series, and S Series Devices

Note Not all E Series, M Series, S Series, and C Series devices support analog triggering. Refer to the specifications for your device to determine if your device supports analog triggering.

Many C Series, E Series, M Series, and S Series devices contain a single analog trigger circuit that you can configure for analog triggering. The analog trigger circuitry is a shared resource for the device, and any of the subsystems can use it. This trigger circuitry supports level and slope triggering with hysteresis as well as analog window triggering. After it is configured, the output of this circuitry appears as the Analog Comparison Event, which can be the source for various triggers and clocks within the analog input, analog output, and counter subsystems.

Sharing an Analog Trigger

Even though the analog trigger is a shared resource, only one analog input or analog output task at a time can configure and reserve it. If you want to share the analog trigger among multiple tasks, configure and reserve it in one task, and use the trigger in subsequent tasks by referring to the source of your trigger, clock, or signal of interest as the Analog Comparison Event. For tasks that support multiple types of analog triggers within the same task, all triggers must share the same configuration settings, or you receive an error. For instance, if you want to use an analog trigger for both your Start and Reference Trigger within an analog input task, the configuration settings for the start and Reference Trigger must be identical.

E Series and S Series Valid Sources for the Analog Trigger

PFI 0

Typically, when configuring an analog trigger, you connect your analog signal to the PFI 0 terminal. Because PFI 0 is the trigger source for both analog and digital signals, NI-DAQmx automatically tristates this terminal when a task exporting a signal on the terminal is not in the committed or running state. This behavior when exporting a signal on PFI 0 differs from typical task-based routing with other PFI lines. It prevents accidental connections of an analog signal directly to digital circuitry, which could permanently damage the device. Also, notice that when connecting an analog signal to PFI 0, the terminal configuration is referenced single-ended.

Even when PFI 0 is not the source of your analog trigger, you cannot use PFI 0 for other digital signal routes because the analog trigger takes over the PFI 0 terminal internal to the device when it is enabled. If you try to use the analog trigger and PFI 0 for digital signals at the same time, you receive a routing error.

Note On NI PXI-6132/6133 devices, you cannot use PFI 0 as the source of an analog trigger, but the analog triggering circuitry still reserves PFI 0 for internal routing.

Analog Input Channel

In addition to PFI 0, analog input tasks can trigger off of one of the analog input channels being sampled. Because E Series devices use a scanning architecture, many restrictions are placed on how you can use an analog trigger when the source is one of the channels you are sampling. When you use an analog Start Trigger, the trigger channel must be the first channel in the channel list. When you use an Analog Reference or Pause Trigger, and the analog channel is the source of the trigger, there can be only one channel in the channel list. If you have more than one channel for Pause or Reference Triggers, you must use PFI 0. Since S Series devices do not use a scanning architecture, none of these restrictions apply. Therefore, for an S Series device, you can use any analog input channel as the source of the trigger regardless of how many channels are being sampled or the order of the trigger channel in the sequence.

Scaling with PFI 0 and Analog Input Channels

Scaling, including custom scales, is not applied if PFI 0 is the trigger source. For instance, you would specify the DAQmx Trigger **analog edge level** attribute/property in volts. However, if you use an analog input channel as the trigger source, you could use scaled units.

M Series Valid Sources for the Analog Trigger

APFI 0 and APFI 1

When configuring an analog trigger, connect your analog signal to either the APFI 0 or APFI 1 terminal and specify APFI 0 or APFI 1 as your trigger source.

Analog Input Channel

In addition to APFI 0 and APFI 1, analog input tasks can trigger off of one of the analog input channels being sampled. Because M Series devices use a scanning architecture, many restrictions are placed on how you can use an analog trigger when the source is one of the channels you are sampling. When you use an Analog Start Trigger, the trigger channel must be the first channel in the channel list. When you use an Analog Reference or Pause Trigger, and the analog channel is the source of the trigger, there can be only one channel in the channel list. If you have more than one channel for Pause or Reference Triggers, you must use APFI 0 or APFI 1.

Scaling with APFI 0, APFI 1, and Analog Input Channels

Scaling, including custom scales, is not applied if APFI 0 or APFI 1 is the trigger source. For instance, you would specify the DAQmx Trigger **analog edge level** attribute/property in volts. However, if you use an analog input channel as the trigger source, you could use scaled units.

Device Calibration and Accuracy of the Analog Trigger

The trigger DACs in the analog trigger circuitry on an E Series, M Series, C Series, or S Series device typically contain four less bits of accuracy than the ADC of the device. No hardware calibration is provided for the analog trigger circuitry. In addition, the propagation delay from when a valid trigger condition is met to when the analog trigger circuitry emits the Analog Comparison Event may have an impact on your measurements if the trigger signal has a high slew rate. If you find these conditions have a noticeable impact on your measurements, you can perform software calibration on the analog trigger circuitry by configuring your task as normal and applying a known signal for your analog trigger. Comparing the observed results against the expected results, you can calculate the necessary offsets to apply in software to fine tune the desired triggering behavior.

C Series Valid Sources for the Analog Trigger

Analog Input Channel

The NI 9205 has no APFI 0 or APFI 1 terminal. Analog input tasks using the NI 9205 can trigger off one of the analog input channels being sampled by the NI 9205. When you use an Analog Start Trigger, the trigger channel must be the first channel from the NI 9205 in the channel list, but channels from other C Series devices can come first. When you use an Analog Reference or Pause Trigger, you can use only one channel from the NI 9205 in the channel list, but you can use channels from other C Series devices. You can combine Analog Start, Reference, and Pause Triggers with different configuration settings by using multiple NI 9205 devices. All analog triggers on the same device must share the same configuration settings.

Device Calibration Considerations

Device calibration consists of verifying the measurement accuracy of a device and adjusting for any measurement error. Verification consists of measuring the performance of the device and comparing these measurements to the published specifications. During calibration, you supply and read voltage levels or other signals using external standards, then you adjust the device calibration constants. The new calibration constants are stored in the EEPROM. These calibration constants are loaded from memory as needed to adjust for the error in the measurements taken by the device.

This section contains information needed for device calibration:

- AO Series
- DSA
- E Series
- M Series, NI 6010, NI 9205, NI 9206
- NI 6154
- NI PXI-6608
- S Series
- SCXI-1600

AO Series Calibration

Your device uses software calibration to fine-tune the analog output circuitry. The software must be programmed (or loaded) with certain numbers called calibration constants. Those constants are stored in nonvolatile memory (EEPROM) on your device or are maintained by NI-DAQmx. To achieve specification accuracy, you should self-calibrate your device just before a measurement session but after your computer and the device have been powered on and warmed up for at least 15 minutes. You should allow this same warm-up time before performing any calibration of your system. Frequent calibration produces the most stable and repeatable measurement performance. The device is not harmed in any way if you recalibrate it often.

Static AO devices, such as the NI 6703 and NI 6704, do not self-calibrate or automatically calibrate. You must use a manual procedure to calibrate static AO devices.

Refer to ni.com/calibration for detailed instructions about calibrating your device.

Note Calibrating your AO device takes some time. Do not be alarmed if the Self-Calibrate or Adjust AO Series Calibration function/VI takes several seconds to execute.

Note For best results, stop any ongoing tasks and disconnect any unnecessary external connections before running calibration.

Calibration Constant Loading by NI-DAQmx

NI-DAQmx automatically loads calibration constants into the software whenever you call functions/VIs that depend on them.

Related Topic

Calibration Signal Connections for AO Series Devices

Virtual Channel Calibration Support

The following devices do not support NI-DAQmx virtual channel calibration:

- NI DAQPad-6015
- NI DAQPad-6016
- NI PCI-6010
- NI PCI-6013
- NI PCI-6014
- NI USB-6008
- NI USB-6009
- SensorDAQ

DSA Calibration

Your device contains digital correction circuitry to compensate for gain and offset errors in the analog and ADC circuitry. The gain and offset calibration constants are stored in nonvolatile memory (EEPROM) on your device. NI-DAQmx writes these calibration constants to the digital correction circuitry.

To achieve the maximum accuracy, you should perform external calibration at least once per year (the recommended external calibration interval) and perform self-calibration prior to measurement sessions or otherwise, as desired. You should calibrate your device only after your computer and the device have been powered on and warmed up for at least 15 minutes.

Self-Calibration

Self-calibration is executed with the Self Calibrate VI/function. When you self-calibrate a DSA device, you do not need signal connections. However, for devices with analog output channels, values generated on those output channels can change during the calibration process. If you have external equipment connected to the AO channels and changing the AO voltage could damage the external equipment, you should disconnect the external equipment before performing the self-calibration.

External Calibration

External calibration is performed using a customized calibration program and external test equipment that has itself been calibrated to the required accuracy or standard. This operation is usually performed by a specialized metrology laboratory. The equipment and connections required to externally calibrate a device varies depending on the device category. For NI PXI-447*X* and PXI-446*X* devices, you need a stable and accurate DC voltage signal to calibrate the AI subsystem. NI PXI-446*X* devices also include an adjustable frequency timebase. You need a stable sinusoidal frequency source to calibrate this timebase. The NI PXI-4461 also supports analog output. You need a digital multimeter (DMM) to calibrate the AO subsystem. The DC voltage, frequency source, and DMM can be manually or automatically controlled and switched between channels, depending on the nature of the customized calibration program.

E Series Calibration

Your device uses hardware calibration to adjust the analog circuitry. This calibration is done with calibration digital-to-analog converters, called calDACs, that fine-tune the analog circuitry. The calDACs must be programmed (or loaded) with certain numbers called calibration constants. Those constants are stored in nonvolatile memory (EEPROM) on your device or are maintained by NI-DAQmx. NI recommends that you self-calibrate your device just before a measurement session but after your computer and the device have been powered on and warmed up for at least 15 minutes. You should allow this same warm-up time before performing any calibration of your system. Frequent calibration produces the most stable and repeatable measurement performance. The device is not harmed in any way if you recalibrate it often.

Note Calibrating your MIO or AI device takes some time. Do not be alarmed if the Self-Calibrate or Adjust E Series Calibration function/VI takes several seconds to execute.

Note For best results, stop any ongoing tasks and disconnect any unnecessary external connections before running calibration.

Calibration Constant Loading by NI-DAQmx

NI-DAQmx automatically loads calibration constants into calDACs whenever you call functions/VIs that depend on them. The following conditions apply:

- 12-bit E Series devices
  - 12-bit devices use a single set of calibration constants for both unipolar and bipolar modes of analog input.
  - One set of constants is valid for unipolar, and another set is valid for bipolar configuration of the analog output channels. When you change the polarity of an analog output channel, NI-DAQmx reloads the calibration constants for that channel.
- 16-bit E Series devices
  - Calibration constants required by the 16-bit E Series devices for unipolar analog input channels are different from those for bipolar analog input channels. If you are acquiring data from one channel, or if all of the channels you are acquiring data from are configured for the same polarity, NI-DAQmx selects the appropriate set of calibration constants for you. If you are scanning several channels, and you mix channels configured for unipolar and bipolar mode in your scan list, NI-DAQmx loads the calibration constants that correspond to the first channel in the scan list.
  - NI 6025E devices use a single set of calibration constants for both unipolar and bipolar modes of analog input.
  - One set of constants is valid for unipolar, and another set is valid for bipolar configuration of the analog output channels. When you change the polarity of an analog output channel, NI-DAQmx reloads the calibration constants for that channel.

Related Topic

Calibration Signal Connections for E Series Devices

M Series, NI 6010, NI 9205, and NI 9206 Calibration

Your device uses software calibration to adjust the software scaling of signals read from and produced by your device. Using calibration pulse width modulated (PWM) sources with a reference voltage, your device measures and calculates scaling constants for analog input and analog output. The scaling constants are stored in nonvolatile memory (EEPROM) on your device. NI recommends that you self-calibrate your device just before a measurement session but after your computer and the device have been powered on and warmed up for at least 15 minutes. You should allow this same warm-up time before performing any calibration of your system. Frequent calibration produces the most stable and repeatable measurement performance. The device is not harmed in any way if you recalibrate it often.

Note Calibrating your device takes some time. Do not be alarmed if the Self-Calibrate or Adjust M Series Calibration function/VI takes several seconds to execute.

Note For best results, stop any ongoing tasks and disconnect any unnecessary external connections before running calibration.

NI 6154 Calibration

Your device uses software calibration to adjust the software scaling of signals read from and produced by your device. Using calibration pulse-width modulated (PWM) sources with a reference voltage, your device measures and calculates scaling constants for analog input and analog output. The scaling constants are stored in nonvolatile memory (EEPROM) on your device. NI recommends that you self-calibrate your device just before a measurement session but after your computer and the device have been powered on and warmed up for at least 15 minutes. You should allow this same warm-up time before performing any calibration of your system. Frequent calibration produces the most stable and repeatable measurement performance. The device is not harmed in any way if you recalibrate it often.

Note Calibrating your device takes some time. Do not be alarmed if the Self-Calibrate or Adjust S Series Calibration function/VI takes several seconds to execute.

Note For best results, stop any ongoing tasks and disconnect any unnecessary external connections before running calibration.

NI PXI-6608 Calibration

You cannot calibrate the PXI-6608 in NI-DAQmx. The device must be calibrated in Traditional NI-DAQ (Legacy). To use the NI PXI-6608 in NI-DAQmx after calibrating it in Traditional NI-DAQ (Legacy), you must do one of the following:

- Call the Traditional NI-DAQ (Legacy) Device Reset function/VI.

  —or—

- Right-click the **Traditional NI-DAQ (Legacy) Devices** folder in MAX and select **Reset Driver for Traditional NI-DAQ**.

S Series Calibration

Your device uses hardware calibration to adjust the analog circuitry. This calibration is done with calibration digital-to-analog converters, called calDACs, that fine-tune the analog circuitry. The calDACs must be programmed (or loaded) with certain numbers called calibration constants. Those constants are stored in nonvolatile memory (EEPROM) on your device or are maintained by NI-DAQmx. NI recommends that you self-calibrate your device just before a measurement session but after your computer and the device have been powered on and warmed up for at least 15 minutes. You should allow this same warm-up time before performing any calibration of your system. Frequent calibration produces the most stable and repeatable measurement performance. The device is not harmed in any way if you recalibrate it often.

Note Calibrating your MIO or AI device takes some time. Do not be alarmed if the DAQmx Self-Calibrate or Adjust S Series Calibration function/VI takes several seconds to execute.

Note For best results, stop any ongoing tasks and disconnect any unnecessary external connections before running calibration.

Calibration Constant Loading by NI-DAQmx

NI-DAQmx automatically loads calibration constants into the calDACs whenever you call functions/VIs that depend on them. The following conditions apply:

- One set of calibration constants is required for each analog input channel on an S Series device. Within this set of calibration channels, different calibration constants are stored for each analog input range supported by the S Series device. When you configure an acquisition, NI-DAQmx automatically loads the calibration constants for each channel, based on its configured input range.
- One set of calibration constants is required for each analog output channel. When an you configure analog output generation, NI-DAQmx automatically loads the calDACs for each channel configured.

Related Topic

Calibration Signal Connections for S Series Devices

SCXI-1600 Calibration

The external calibration process for the SCXI-1600 module is nearly identical to the E Series devices. However, when applying a precision voltage to the module, you must connect the signal to the EXTCAL BNC connector on the front of the SCXI-1600, instead of the AI0 analog input channel. In LabVIEW, use the Adjust E-Series Calibration VI, instead of the Adjust SC Baseboard Calibration VI.

Signal Connections

This section contains information about calibration signal connections for AO Series, E Series, M Series, NI 6010, NI 6154, and S Series devices.

Device Calibration Signal Connections for AO Series Devices

Self-Calibration

When you self-calibrate your AO Series device, no signal connections are necessary. However, values generated on the analog output channels change during the calibration process.

External Calibration

When externally calibrating your AO Series device, connect the signals as described below for the type of AO Series device you are calibrating. Set the reference voltage between +6.000 V and +9.999 V. Typically, you use a calibrator or other stable voltage source for the reference voltage. Do not use a power supply as its signals are not very stable.

Follow these steps for AO Series devices:

1. Connect the positive output of your reference voltage source to the EXT REF terminal.
2. Connect the negative output of your reference voltage source to the AO GND terminal.

For more information about calibration procedures for static AO devices, refer to the device documentation or to Calibration Procedures.

Device Calibration Signal Connections for E Series Devices

Self-Calibration

When you self-calibrate your E Series device, no signal connections are necessary. However, values generated on the analog output channels change during the calibration process. If you have external circuitry connected to the analog output channels and you do not want changes on these channels, you should disconnect the circuitry before beginning the self-calibration.

External Calibration

When externally calibrating your E Series device, connect the signals as described below for the type of E Series device you are calibrating. Set the reference voltage between +6.000 V and +9.999 V. Typically, you use a calibrator or other stable voltage source for the reference voltage. Do not use a power supply as its signals are not very stable.

12-Bit E Series Devices

Follow these steps for 12-bit E Series devices:

1. Connect the positive output of your reference voltage source to physical channel ai8.
2. Connect the negative output of your reference voltage source to the AI SENSE terminal.
3. Connect physical channel ao0 to physical channel ai0.
4. If your reference voltage source and your computer are floating with respect to each other, connect the AI SENSE terminal to the AI GND terminal as well as to the negative output of your reference voltage source.

16-Bit E Series Devices

Follow these steps for 16-bit E Series devices:

1. Connect the positive output of your reference voltage source to physical channel ai0.
2. Connect the negative output of your reference voltage source to physical channel ai8.
3. If your reference voltage source and your computer are floating with respect to each other, connect the negative output of your reference voltage source to the AI GND terminal as well as to physical channel ai8.

How Does Calibration in NI-DAQmx Differ from Calibration in Traditional NI-DAQ (Legacy)?

In Traditional NI-DAQ (Legacy), several 16-bit devices use the 12-bit signal connection scheme (the NI PCI-6034, NI PCI-6035, NI 6036 and NI PCI/PXI 6052). In NI-DAQmx, all of these devices now use the 16-bit E Series connections.

Device Calibration Signal Connections for S Series Devices

Self-Calibration

When you self-calibrate your S Series device, no signal connections are necessary. However, values generated on the analog output channels change during the calibration process. If you have external circuitry connected to the analog output channels that is sensitive to these changes, you should disconnect the circuitry before beginning self-calibration.

External Calibration

When externally calibrating your S Series device, connect the signals as described below. Set the reference voltage to the following:

- NI PCI/PXI 6143: between 3.0 V and 4.998 V
- NI PCI/PXI 6115: between 4.995 V and 5.005 V
- NI PCI/PXI 6120: between 4.995 V and 5.005 V
- All other S Series Devices: between 6.0 V and 9.998 V

Typically, you should use a calibrator or other stable voltage source for calibration. Do not use a power supply as its signals are not very stable.

For external calibration, make the following signal connections:

1. Connect the positive output of your reference voltage source to ACH0+.
2. Connect the negative output of your reference voltage source to ACH0-.

Device Calibration Signal Connections for M Series and NI 6010 Devices

Self-Calibration

When you self-calibrate your M Series or NI 6010 device, no signal connections are necessary. However, values generated on the analog output channels change during the calibration process. If you have external circuitry connected to the analog output channels and you do not want changes on these channels, you should disconnect the circuitry before beginning the self-calibration.

External Calibration

When externally calibrating your M Series or NI 6010 device, connect the signals as described below for the type of device you are calibrating. Set the reference voltage between +6.0 V and +8.5 V for M Series devices, between +3.5 V and 4.0 V for NI 6010 devices. Typically, you use a calibrator or other stable voltage source for the reference voltage. Do not use a power supply as its signals are not very stable.

Follow these steps:

1. Disconnect any external connections or circuitry to your device.
2. Connect the positive output of your reference voltage source to physical channel ai0.
3. Connect the negative output of your reference voltage source to physical channel ai8.
4. If your reference voltage source and your computer are floating with respect to each other, connect the negative output of your reference voltage source to the AI GND terminal as well as to physical channel ai8.

Device Calibration Signal Connections for the NI 6154

Self-Calibration

When you self-calibrate your NI 6154 device, no signal connections are necessary. However, values generated on the analog output channels change during the calibration process.

External Calibration

When externally calibrating your NI 6154 device, connect the signals as described below for the type of device you are calibrating. Set the reference voltage between +6.0 V and +9.998 V. Typically, you use a calibrator or other stable voltage source for the reference voltage. Do not use a power supply as its signals are not very stable.

Follow these steps:

1. Disconnect any external connections or circuitry to your device.
2. Connect the positive output of your reference voltage source to physical channel AI0+.
3. Connect the negative output of your reference voltage source to physical channel AI0-.

Counters

This section contains information on counter signal connections and block diagrams that illustrate the internal counter routing.

Connecting Counter Signals

The default terminals used for counter measurements and generations vary from device to device. Follow the links below for information specific to your device. To override the default input terminal, set the DAQmx Channel **Input Terminal** attribute/property for the measurement type. For instance, if you are counting edges, you would use **Counter Input:Count Edges:Input Terminal**. To override the default output terminal, set the DAQmx Channel **Output Terminal** attribute/property to the desired value.

Related Topics

- AO Series, E Series, and S Series Signal Connections for Counters
- Bus-Powered M Series Signal Connections for Counters
- C Series Signal Connections for Counters
- M Series Signal Connections for Counters
- 37-Pin DSUB Signal Connections for Counters
- TIO Signal Connections for Counters

Bus-Powered M Series Signal Connections for Counters

The following table lists the default input terminals for various counter measurements on bus-powered M 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.

16-PFI Line Devices (NI 6218)

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

16-PFI Line Devices (NI 6212/6216)

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

8-PFI Line Devices (Such as the NI 6210/6211/6215)

Measurement	Ctr0	Ctr1
Count Edges	Edges: PFI 0 Count Direction: PFI 0	Edges: PFI 3 Count Direction: PFI 3
Pulse Width Measurement	PFI 1	PFI 2
Period/Frequency Measurement (Low Frequency with One Counter)	PFI 1	PFI 2
Period/Frequency Measurement (High Frequency with Two Counters)	PFI 0	PFI 3
Period/Frequency Measurement (Large Range with Two Counters)	PFI 0	PFI 3
Semiperiod Measurement	PFI 1	PFI 2
Two-Edge Separation Measurement	Start: PFI 0 Stop: PFI 1	Start: PFI 3 Stop: PFI 2

Position Measurement	A: PFI 0 B: PFI 1 Z: PFI 2	A: PFI 3 B: PFI 2 Z: PFI 1

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 4	PFI 5

C Series Signal Connections for Counters

The following table lists the default input terminals for various counter measurements. 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.

NI 9401, NI 9421, NI 9422, and NI 9423

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

NI 9402

Measurement	Ctr0	Ctr1
Count Edges	Edges: PFI 0 Count Direction: PFI 2	Edges: PFI 3 Count Direction: PFI 1
Pulse Width Measurement	PFI 1	PFI 2
Period/Frequency Measurement (Low Frequency with One Counter)	PFI 1	PFI 2
Period/Frequency Measurement (High Frequency with Two Counters)	PFI 0	PFI 3
Period/Frequency Measurement (Large Range with Two Counters)	PFI 0	PFI 3
Semiperiod Measurement	PFI 1	PFI 2
Two-Edge Separation Measurement	Start: PFI 2 Stop: PFI 1	Start: PFI 1 Stop: PFI 2
Position Measurement	A: PFI 0 B: PFI 2 Z: PFI 1	A: PFI 3 B: PFI 1 Z: PFI 2

NI 9411

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

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

NI 9401, NI 9472, NI 9474, and NI 9485

Ctr0	Ctr1
PFI 3	PFI 7

NI 9481 and NI 9402

Ctr0	Ctr1
PFI 0	PFI 3

AO Series, E Series, and S Series Signal Connections for Counters

The following table lists the default input terminals for various counter measurements. You can use a different PFI line for any of the input terminals, with the exception of the count direction terminal for edge counting. 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[1]: port0/line6	Edges: PFI 3 Count Direction[1]: port0/line7
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
[1] The count direction terminal must be tristated to use an external signal. Reset the device to ensure the terminal is tristated.		

The following table lists the default output terminals for counter output. You must use the default output terminal, with the exception that for Ctr0, you can select a RTSI line.

Ctr0	Ctr1
CTR 0 OUT	CTR 1 OUT

TIO Signal Connections for Counters

The following table lists the default input terminals for various counter measurements. 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 properties/attributes.

Measurement	Ctr0	Ctr1	Ctr2	Ctr3	Ctr4	
Count Edges	Edges: PFI 39 Count Direction: PFI 37	Edges: PFI 35 Count Direction: PFI 33	Edges: PFI 31 Count Direction: PFI 29	Edges: PFI 27 Count Direction: PFI 25	Edges: PFI 23 Count Direction: PFI 21	Ed PF Co Di PF
Pulse Width Measurement	PFI 38	PFI 34	PFI 30	PFI 26	PFI 22	PF
Period/Frequency Measurement (Low Frequency with One Counter)	PFI 38	PFI 34	PFI 30	PFI 26	PFI 22	PF
Period/Frequency Measurement (High Frequency with Two Counters)	PFI 39	PFI 35	PFI 31	PFI 27	PFI 23	PF
Period/Frequency Measurement (Large Range with Two Counters)	PFI 39	PFI 35	PFI 31	PFI 27	PFI 23	PF
Semiperiod Measurement	PFI 38	PFI 34	PFI 30	PFI 26	PFI 22	PF
Two-Edge Separation Measurement	Start: PFI 37	Start: PFI 33	Start: PFI 29	Start: PFI 25	Start: PFI 21	St PF

	Stop: PFI 38	Stop: PFI 34	Stop: PFI 30	Stop: PFI 26	Stop: PFI 22	St PF
Position Measurement	A: PFI 39 B: PFI 37 Z: PFI 38	A: PFI 35 B: PFI 33 Z: PFI 34	A: PFI 31 B: PFI 29 Z: PFI 30	A: PFI 27 B: PFI 25 Z: PFI 26	A: PFI 23 B: PFI 21 Z: PFI 22	A: B: Z:
GPS Timestamp Measurement	N/A	N/A	N/A	N/A	N/A	N/

Note The NI 6601 has only four counters (ctr0–ctr3). The entries in the previous table for cntr4, cntr5, cntr6, and cntr7 do not apply for that device.

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	Ctr4	Ctr5	Ctr6	Ctr7
PFI 36	PFI 32	PFI 28	PFI 24	PFI 20	PFI 16	PFI 12	PFI 8

Note The NI 6601 has only four counters (ctr0–ctr3). The entries in the previous table for cntr4, cntr5, cntr6, and cntr7 do not apply for that device.

NI 6624 Issues

The eight PFI lines listed as the defaults for counter output are dedicated for output, and they are the only terminals you can use for counter output. For example, you can use PFI 8 (the default for Ctr7) as the output terminal for any counter, but you cannot use it as an input terminal.

When using counter output, if the **Idle State** attribute/property is low, the optocouplers on the NI 6624 will still be driving your output load. Set the **Idle State** attribute/property to high to prevent driving the output after your task completes.

37-Pin DSUB Signal Connections for Counters

The following table lists the default input terminals for various counter measurements for devices that use the 37-Pin DSUB connector such as the NI 6010, NI 6154, and NI 623*X*. 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.

Input

Measurement	Ctr0	Ctr1
Count Edges	Edges: PFI 0 Count Direction: PFI 2	Edges: PFI 3 Count Direction: PFI 5
Pulse Width Measurement	PFI 1	PFI 4
Period/Frequency Measurement (Low Frequency with One Counter)	PFI 1	PFI 4
Period/Frequency Measurement (High Frequency with Two Counters)	PFI 0	PFI 3
Period/Frequency Measurement (Large Range with Two Counters)	PFI 0	PFI 3
Semiperiod Measurement	PFI 1	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.

Output

Ctr0	Ctr1
PFI 6	PFI 7

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

Note Some M Series devices, including the NI 6010, NI 6154, NI 6221 (37-pin), and NI 623*X*, use the 37-pin DSUB connector. These devices have different counter terminal defaults. Refer to

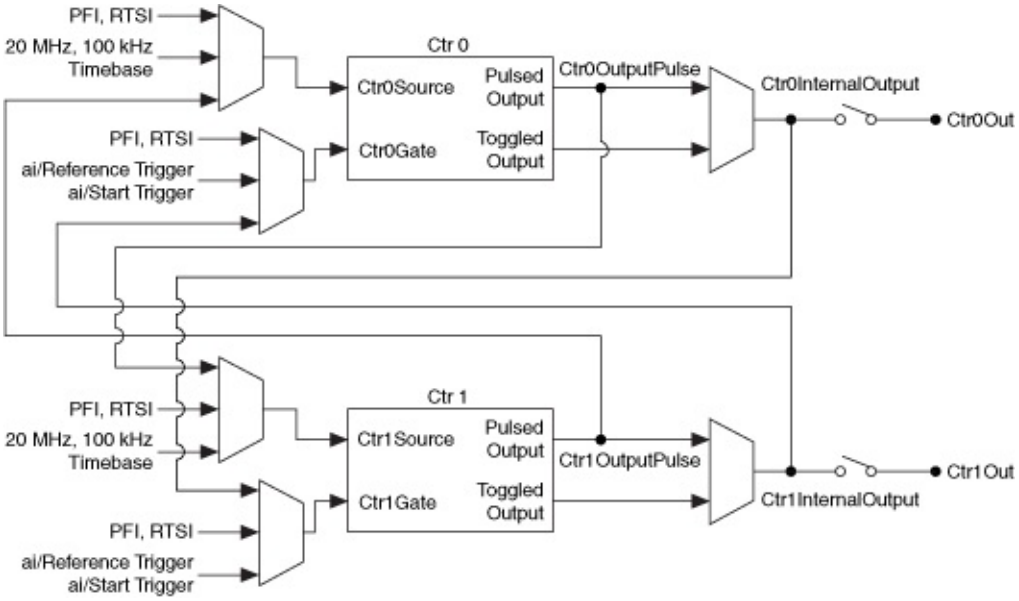
the 37-Pin DSUB Signal Connections for Counters for the default input terminals on these devices. Bus-powered M Series devices, such as the NI USB-621*X* devices, also have different counter terminal defaults. Refer to the Bus-Powered M Series Signal Connections for Counters for the default input terminals on these devices.

Counter Internal Routing Diagrams

This section contains block diagrams that illustrate the internal counter routing for AO Series, C Series, E Series, M Series, S Series, and TIO devices.

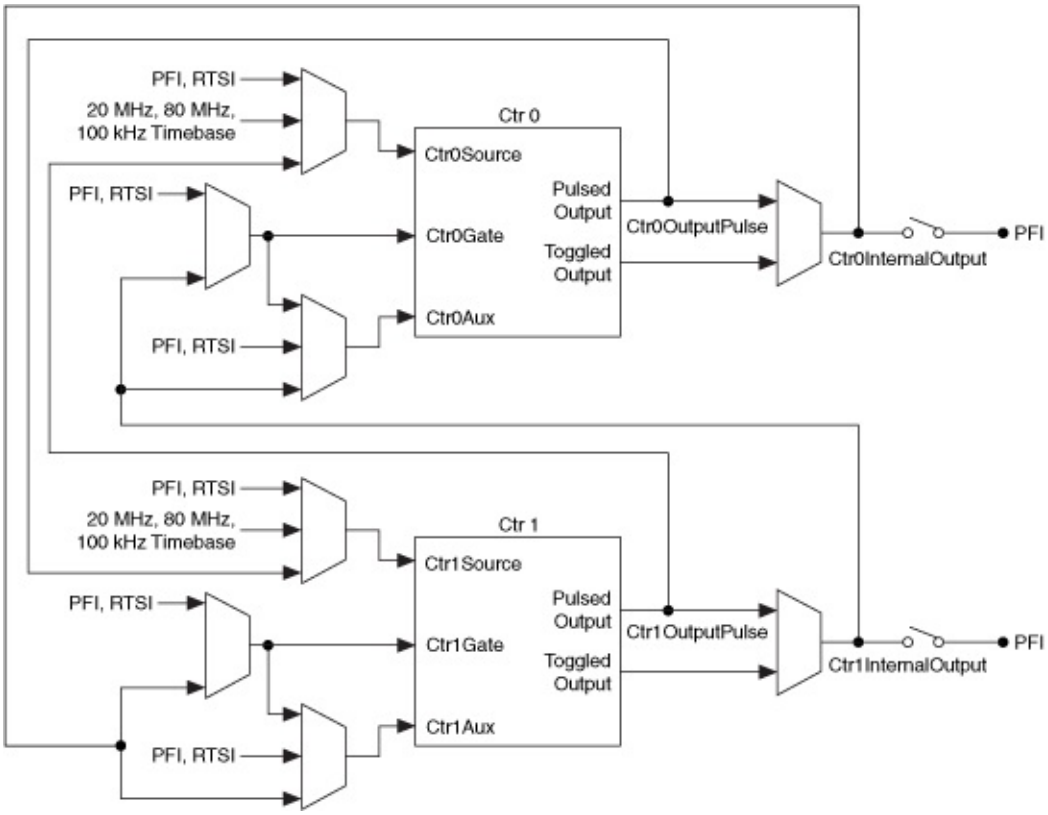
AO Series, E Series, S Series Counter Internal Routing Diagram

The following figure shows the internal routing for DAQ devices with the STC counter/timer such as E Series devices. The black circles represent terminals.



C Series, M Series, and TIO Counter Internal Routing Diagram

The following figure shows the routing for DAQ devices that use the TIO counter/timer such as 66*XX* devices or for devices that use the STC2 timer, which is on M Series devices and the CompactDAQ chassis.



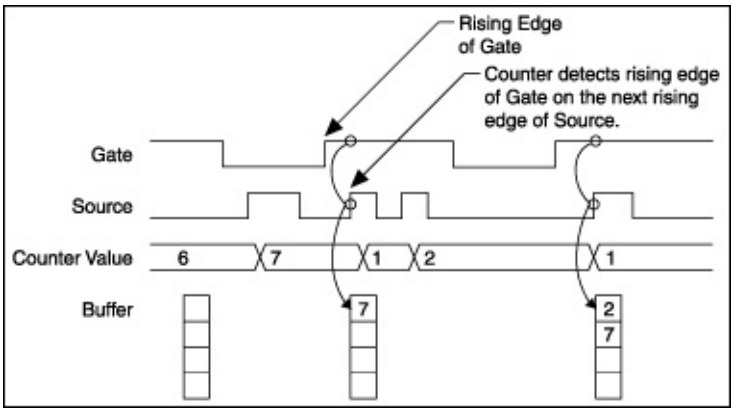
You can use any PFI line on your device. However, there are preferred PFI lines that do not require extra internal resources when used. Refer to Routing Considerations for Timing I/O Devices for additional information.

Duplicate Count Prevention

Duplicate count prevention (or synchronous counting mode) ensures that a counter returns correct data in applications that use a slow or non-periodic external source. Duplicate count prevention applies to any counter application such as measuring frequency or period. In such applications, the counter should store the number of times an external Source pulses between rising edges on the Gate signal.

Example Application That Works Correctly (No Duplicate Counting)

The following figure shows a buffered period measurement that uses an external signal as the Source.

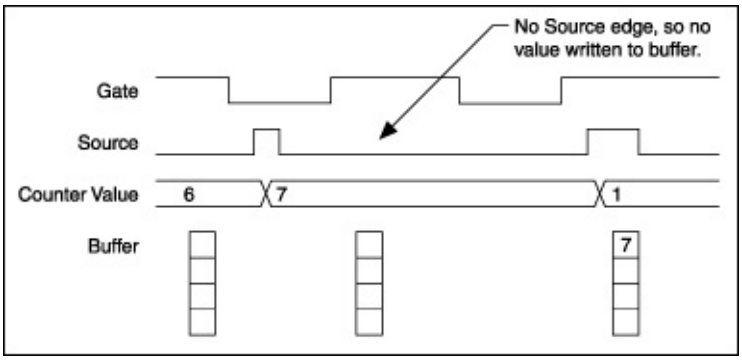


On the first rising edge of the Gate, the current count of 7 is stored. On the next rising edge of the Gate, the counter stores a 2 since two Source pulses occurred after the previous rising edge of Gate.

The counter synchronizes or samples the Gate signal with the Source signal. So the counter does not detect a rising edge in the Gate until the next Source pulse. In this example, the counter stores the values in the buffer on the first rising Source edge after the rising edge of Gate.

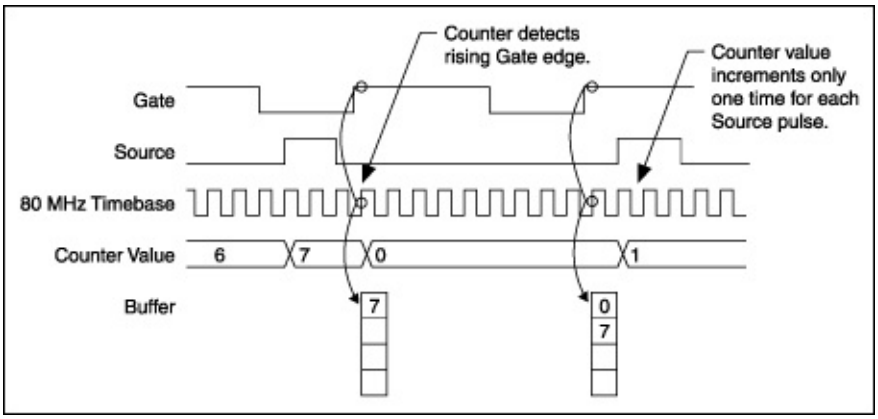
Example Application That Works Incorrectly (Duplicate Counting)

In the following figure, after the first rising edge of Gate, no Source pulses occur. So the counter does not write the correct data to the buffer.



Example Application That Prevents Duplicate Counting

With duplicate count prevention enabled, the counter synchronizes both the Source and Gate signals to the maximum timebase. By synchronizing to the timebase, the counter detects edges on the Gate even if the Source does not pulse. This enables the correct current count to be stored in the buffer even if no Source edges occur in between Gate signals. Refer to the following example.



Even if the Source pulses are long, the counter increments only once for each Source pulse.

Normally, the counter value and Counter *n* Internal Output signals change synchronously to the Source signal. With duplicate count prevention, the counter value and Counter *n* Internal Output signals change synchronously to the maximum timebase.

When To Use Duplicate Count Prevention

You should use duplicate count prevention if the following conditions are true.

- You are making a counter measurement
- You are using an external signal (such as PFI *x*) as the counter Source
- The frequency of the external Source is 25% of your maximum timebase or less
- You can have counter value and output to change synchronously with the maximum timebase

In all other cases, you should *not* use duplicate count prevention.

Enabling and Disabling Duplicate Count Prevention in NI-DAQmx

NI-DAQmx enables duplicate count prevention by default except in the following cases:

- The input terminal is an onboard timebase.
- Prescaling is enabled.
- The timing type is on demand.
- The **Counter Output Event Output Terminal** attribute/property is used in your application.

You can enable and disable duplicate count prevention in NI-DAQmx with the **Enable Duplicate Count Prevention** attribute/property.

Prescaling

Prescaling allows the counter to count a signal that is faster than the maximum timebase of the counter. The TIO counters offer 8X and 2X prescaling on each counter. You can disable prescaling. Each prescaler consists of a small, simple counter that counts to eight (or two) and rolls over. This counter is specifically designed for this application and can count signals that are faster than the general purpose counters. The Ctr*N*source signal on the general purpose counter is the divided signal from the simple counter.

Prescaling is for two-counter period and frequency measurements in which the measurement is made on a continuous, repetitive signal. The prescaling counter cannot be read; therefore, you cannot determine how many edges have occurred since the previous roll-over. Prescaling can be also used for counting edges provided it is acceptable to have an error of up to seven when using 8X prescaling or one when using 2X prescaling.

Counter Input Error Reporting with M Series USB and C Series

With C Series and M Series USB devices (except bus-powered M Series devices), buffered counter input error reporting occurs every 128 samples for high-speed USB and every 16 samples for full-speed USB. When an error is detected, the task stops. To prevent the acquisition of incorrect samples, only data within either the 128- or 16-sample group is reported. For instance, if you attempt to acquire 256 samples, and an error occurs after sample 250, the task returns 128 samples instead of 249. If you attempt to acquire a number of samples less than or equal to 128 or 16 and an error occurs, the task returns no samples.

Digital Filtering

Digital filtering rejects state transitions that do not stay at a state for a specified amount of time. For example, for an edge counting measurement with digital filtering, the device does not count an edge if the pulse width is not at least the specified time. For digital input tasks, the device does not recognize that a signal changed from one state to another unless the signal remains at that state for the specified amount of time.

This section contains information about digital filtering for C Series, DIO, M Series, and TIO devices.

Digital Filtering Considerations for C Series and M Series Devices

Digital debouncing filters are only supported on counter inputs. Each PFI line can independently select from three fixed values (125 ns, 6.425 µs, 2.55 ms). For each counter input attribute/property, there are two attributes/properties associated with digital debounce filtering: **Digital Filter Enable** and **Digital Filter Minimum Pulse Width**.

When you set the **Digital Filter Enable** to true, you must also configure the **Digital Filter Minimum Pulse Width** attribute/property. This value represents the minimum value that is guaranteed to pass into the STC II. Refer to your device documentation to determine the minimum pulse width guaranteed to be blocked.

The following table lists the counter input terminals that can be digitally filtered.

Type	**Attribute/Property**
Channel	Frequency Input Terminal
	Period Input Terminal
	Count Edges Input Terminal
	Count Edges Count Direction
	Position A Input Terminal
	Position B Input Terminal
	Position Z Input Terminal
	Pulse Width Input Terminal
	Two-Edge First Input Terminal
	Two-Edge Second Input Terminal
	Semi-Period Input Terminal
	Counter Input Timebase Source
	Counter Output Timebase Source
Timing	Sample Clock Source
Triggering	Start Trigger Source
	Pause Trigger Source

	Arm Start Trigger Source

Digital Filtering Considerations for DIO Devices

Digital filtering is enabled by default on all isolated DIO devices that support digital filtering. The default minimum pulse width is 0.1 ms or 100 µs. Refer to the following table for a list of devices and their digital filtering settings.

Digital Filtering Setting	Devices
Digital Filtering Enabled by Default	NI PCI-6510 NI PCI-6511 NI PCI-6514 NI PCI-6515 NI PCI-6518 NI PCI-6519 NI PCI-6527 NI PCI-6528 NI PXI-6511 NI PXI-6514 NI PXI-6515 NI PXI-6527 NI PXI-6528 NI PXI-6529
Digital Filtering Disabled by Default	NI PCI-6509 NI PXI-6509

Digital Filtering Not Supported	NI PCI-6503
	NI PCI-6512
	NI PCI-6513
	NI PCI-6516
	NI PCI-6517
	NI PCI-DIO-96
	NI PXI-6508
	NI PXI-6512
	NI PXI-6513
	NI USB-6501

Digital Filtering Considerations for TIO-Based Devices

There are two methods for filtering and synchronizing digital signals. One method is to synchronize the input signal to the maximum onboard timebase on the device. To do this, set **Digital Synchronization Enable** to true.

The other method is to pass the input of any PFI line through a digital debouncing filter. Each PFI line can independently select from four fixed values (5 µs, 1 µs, 500 ns, 100 ns) and one custom filter value. The custom filter value must be the same for each PFI line. That is, if you choose a filter value of 2 µs for a PFI line, other PFI lines on the device at the same time can only choose from the four fixed values and the 2 µs value selected as the custom filter value. For each counter input property, there are four attributes/properties associated with digital debounce filtering: **Digital Filter Enable**, **Digital Filter Minimum Pulse Width**, **Digital Filter Timebase Source**, and **Digital Filter Timebase Rate**.

When you set the **Digital Filter Enable** to true, you must also configure the **Digital Filter Minimum Pulse Width** attribute/property. This value represents the minimum value that is guaranteed to pass into the TIO. The minimum pulse width guaranteed to be blocked is one-half of the **Digital Filter Minimum Pulse Width** attribute/property. When you select a custom filter value with the **Minimum Pulse Width** attribute/property, NI-DAQmx uses an internal 32-bit utility counter to generate the desired filter value. If you would like to generate the filter clock using your own external signal, you can use the **Digital Filter Timebase Source** and **Digital Filter Timebase Rate** attributes/properties. You must configure both to use an external signal as the source for the digital filter.

You cannot set both **Digital Filter Enable** and **Digital Synchronization Enable** to true at the same time. You can use only one of these digital filtering methods at a time.

The following table lists the counter input terminals that can be digitally filtered.

Type	**Attribute/Property**
Channel	Frequency Input Terminal
	Period Input Terminal

	Count Edges Input Terminal
	Count Edges Count Direction
	Position A Input Terminal
	Position B Input Terminal
	Position Z Input Terminal
	Pulse Width Input Terminal
	Two-Edge First Input Terminal
	Two-Edge Second Input Terminal
	Semi-Period Input Terminal
	Counter Input Timebase Source
	Counter Output Timebase Source
Timing	Sample Clock Source
Triggering	Start Trigger Source
	Pause Trigger Source
	Arm Start Trigger Source

Digital I/O

This section contains information specific to DIO devices.

Change Detection

This section contains information about change detection for C Series, DIO, and M Series devices.

Change Detection Considerations for NI 6527 Devices

The **Change Detection Overflowed** attribute/property uses the change detection overflow circuitry on a DIO device to determine if an overflow occurred. The NI 6527 change detection overflow circuitry does not detect an overflow if a single rising edge and a single falling edge are detected prior to reading a sample. It will detect overflows if two rising edges or two falling edges occur prior to reading a sample.

Change-Detection Considerations for C Series and M Series Devices

When performing a buffered change-detection task with an M Series device or a CompactDAQ system, the correlated digital input circuitry is automatically reserved and used for the task. Non-buffered tasks, including hardware-timed single point, do not reserve or use the correlated digital input circuitry.

Digital I/O Considerations for C Series Devices

Digital I/O module capabilities depend on the type of digital signals that the module can measure or generate. Static digital I/O modules are designed for signals that change slowly and are accessed by software-timed reads and writes. Static digital I/O modules might take longer to update than their specifications indicate. Correlated digital I/O modules are for signals that change rapidly and are updated by either software or hardware-timed reads and writes. In addition, correlated digital I/O modules can perform the following tasks:

- Hardware-timed digital input/output tasks (when used in slots 1 through 4)
- Counter/timer tasks (when used in slots 5 and 6)
- Accessing PFI signal tasks (when used in slots 5 and 6)

The NI 9403, NI 9425, NI 9476, and NI 9477 are static digital I/O modules. The NI 9401, NI 9402, NI 9411, NI 9421, NI 9422, NI 9423, NI 9435, NI 9472, NI 9474, NI 9481, and NI 9485 are correlated digital I/O modules.

Sample Clock Timing for Digital I/O

You can use sample clock timing for digital I/O on the following devices.

AO Series[1]

- NI 673*X*

C Series[1]

Correlated digital I/O devices support sample clock timing as long as you use slots 1, 2, 3, or 4. Refer to Digital I/O Considerations for C Series for more information.

M Series[1]

- NI 622*X*
- NI 625*X*
- NI 628*X*

S Series[1]

- NI 6115
- NI 6120
- NI 6132
- NI 6133

NI 653*X*

- NI PCI-6533 (DIO-32HS)
- NI PCI-6534
- NI PXI-6533
- NI PXI-6534

[1]There is no dedicated onboard sample clock for digital I/O on these devices. You must use a different clock, typically the AI or AO Sample Clock.

Handshake Timing Devices

You can use handshake timing for digital I/O on the following devices:

- NI PCI-6025E
- NI PCI-6533 (DIO-32HS)
- NI PCI-6534
- NI PCI-DIO-24
- NI PCI-DIO-96
- NI PXI-6025E
- NI PXI-6508
- NI PXI-6533
- NI PXI-6534
- NI PCIe-6536
- NI PCIe-6537

Burst Handshaking Timing Defaults for NI 653X Devices

The following table lists the default terminals used for burst handshake timing.

Note The NI 6533 and NI 6534 have two timing engines, Timing Engine 1 and Timing Engine 0. Each timing engine is associated with PFI lines. The timing engine you use is determined by the digital lines you use. If the least significant port is port 0, NI-DAQmx picks Timing Engine 0. If the least significant port is port 2, NI-DAQmx picks Timing Engine 1.

Device	Pause Trigger Default	Ready for Transfer Event Default
NI PCI-6533 (DIO-32HS)	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PCI-6534	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PXI-6533	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PXI-6534	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PCIe-6536	PFI 0	PFI 1
NI PCIe-6537	PFI 0	PFI 1

The recommended sample clock terminal for burst handshake timing is PFI 4 (Timing Engine 0) or PFI 5 (Timing Engine 1).

Burst Handshake Timing for Digital I/O

You can use burst handshake timing for digital I/O on the following devices:

- NI PCI-6533 (DIO-32HS)
- NI PCI-6534
- NI PXI-6533
- NI PXI-6534
- NI PCIe-6536
- NI PCIe-6537

Handshake Timing Defaults

The following table lists the default terminals used for handshake timing for NI 653*X* devices.

Note The NI 6533 and NI 6534 have two timing engines, Timing Engine 1 and Timing Engine 0. Each timing engine is associated with PFI lines. The timing engine you use is determined by the digital lines you use. If the least significant port is port 0, NI-DAQmx picks Timing Engine 0. If the least significant port is port 2, NI-DAQmx picks Timing Engine 1.

Device	Handshake Trigger Default	Handshake Event Default
NI PCI-6533 (DIO-32HS)	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PCI-6534	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PXI-6533	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PXI-6534	PFI 2 (Timing Engine 0), PFI 3 (Timing Engine 1)	PFI 6 (Timing Engine 0), PFI 7 (Timing Engine 1)
NI PCIe-6536	PFI 0	PFI 1
NI PCIe-6537	PFI 0	PFI 1

The recommended sample clock terminal for burst handshake timing is PFI 4 (Timing Engine 0) or PFI 5 (Timing Engine 1).

Watchdog Timers

Watchdog timers are a hardware feature that you can use to detect a failure in the software controlling the device. Software failures could include a system crash or a loop rate that is slower than you intend. To use a watchdog timer, you must use a watchdog timer task. When you create a watchdog timer task, you specify the timeout value for the watchdog timer and a set of expiration states for digital output physical channels on the device. The channels go to those expiration states if the watchdog timer expires.

Your application must continuously reset the watchdog timer to prevent it from expiring. For example, if you have a digital I/O application, and you expect a loop in the application to acquire and analyze data 10 times per second, you should set the watchdog timer to expire in 100 ms and reset the timer inside the digital I/O loop. If the loop does not execute once every 100 ms, the watchdog timer expires and the device goes into the expired state. You must then clear the expiration or reset the device.

Also, you can use the Expiration Trigger to cause the watchdog timer to expire. Set the timeout of the watchdog timer task to -1 to disable expiration due to timeout if you want the Expiration Trigger to be the only mechanism to cause expiration.

Pause Triggering

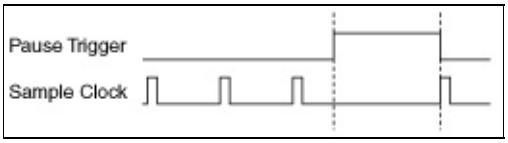
This section contains information about Pause Triggering for AO Series, DSA, E Series, M Series, S Series, and TIO devices.

Pause Trigger Considerations for AO Series Devices

The source of your sample clock can affect when your generation resumes after the deassertion of a Pause Trigger.

Analog Output

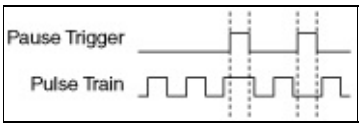
When you generate analog output signals, the generation pauses as soon as the Pause Trigger is asserted. If the source of your sample clock is the onboard clock, the generation resumes as soon as the Pause Trigger is deasserted.



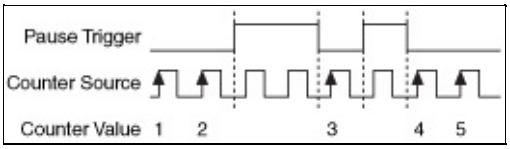
If you are using any signal other than the onboard clock as the source of your sample clock, the generation resumes as soon as the Pause Trigger is deasserted and another edge of the sample clock is received, as shown in the following figure.

Counters

Continuous pulse-train generation: The pulse-train generation pauses as soon as the Pause Trigger is asserted, not at the end of a pulse. The pulse train resumes after the Pause Trigger is deasserted. A Pause Trigger elongates either the high or low pulse depending on which one was being generated at the time the Pause Trigger was asserted.



Nonbuffered edge counting: The counter stops counting edges as soon as the Pause Trigger is asserted and resumes counting edges after the Pause Trigger is deasserted.



Pause Trigger Considerations for DSA Devices

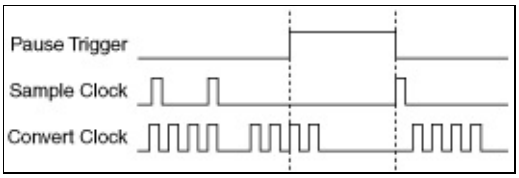
DSA devices do not support Pause Triggering.

Pause Trigger Considerations for E Series and M Series Devices

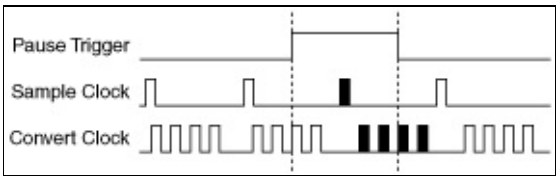
The source of your sample clock often can affect when your acquisition or generation pauses and resumes with the assertion and deassertion of a Pause Trigger.

Analog Input

When you measure analog input signals and the Pause Trigger is asserted, the current sample across all channels finishes before pausing. For instance, if you are sampling four channels and the second channel is being sampled at the time the Pause Trigger is asserted, the second, third, and fourth channels complete their sample before the acquisition pauses. If you are using the onboard clock as the source of your sample clock, the acquisition resumes as soon as the Pause Trigger is deasserted.

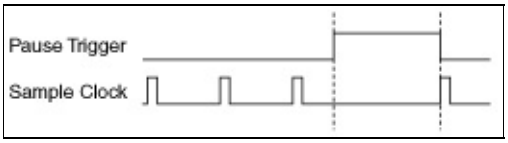


If you are using any signal other than the onboard clock as the source of your sample clock, the acquisition resumes as soon as the Pause Trigger is deasserted and another edge of the sample clock is received as shown in the following figure.



Analog Output

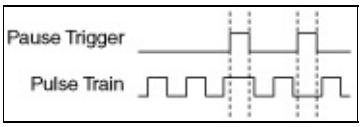
When you generate analog output signals, the generation pauses as soon as the Pause Trigger is asserted. If the source of your sample clock is the onboard clock, the generation resumes as soon as the Pause Trigger is deasserted.



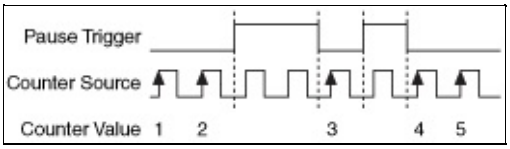
If you are using any signal other than the onboard clock as the source of your sample clock, the generation resumes as soon as the Pause Trigger is deasserted and another edge of the sample clock is received as shown in the following figure.

Counters

Continuous pulse-train generation: The pulse-train generation pauses as soon as the Pause Trigger is asserted, not at the end of a pulse. The pulse train resumes after the Pause Trigger is deasserted. A Pause Trigger elongates either the high or low pulse depending on which one was being generated at the time the Pause Trigger was asserted.



Nonbuffered edge counting: The counter stops counting edges as soon as the Pause Trigger is asserted and resumes counting edges after the Pause Trigger is deasserted.

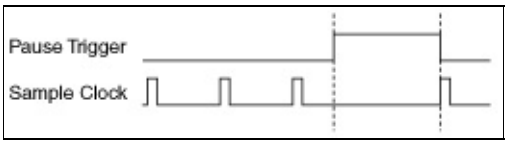


Pause Trigger Considerations for S Series Devices

The source of your sample clock often can affect when your acquisition or generation pauses and resumes with the assertion and deassertion of a Pause Trigger.

Analog Input and Analog Output

When you generate analog output signals or acquire analog input signals, the generation/acquisition pauses as soon as the Pause Trigger is asserted. If the source of your sample clock is the onboard clock, the generation/acquisition resumes as soon as the Pause Trigger is deasserted.

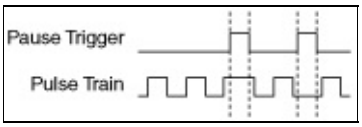


If you are using any signal other than the onboard clock as the source of your sample clock, the generation/acquisition resumes as soon as the Pause Trigger is deasserted and another edge of the sample clock is received as shown in the following figure.

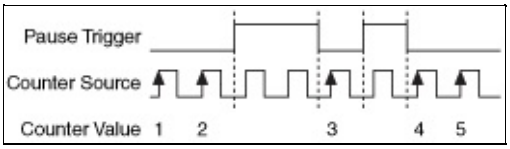
Pause triggers also require special consideration when used on a device with a pipelined ADC. See the S Series Timing Considerations for how pipelined ADCs and Pause Triggers can affect your measurement.

Counters

Continuous pulse-train generation: The pulse-train generation pauses as soon as the Pause Trigger is asserted, not at the end of a pulse. The pulse train resumes after the Pause Trigger is deasserted. A Pause Trigger elongates either the high or low pulse depending on which one was being generated at the time the Pause Trigger was asserted.



Nonbuffered edge counting: The counter stops counting edges as soon as the Pause Trigger is asserted and resumes counting edges after the Pause Trigger is deasserted.



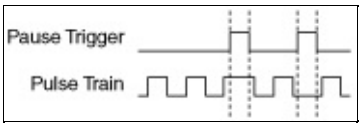
Note The NI 6154 does not support pause triggering.

Pause Trigger Considerations for TIO Devices

Counters

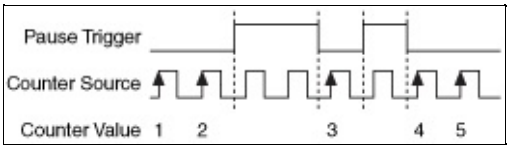
Continuous Pulse-Train Generation

The pulse-train generation pauses as soon as the Pause Trigger is asserted, not at the end of a pulse. The pulse train resumes after the Pause Trigger is deasserted. A Pause Trigger elongates either the high or low pulse depending on which one was being generated at the time the Pause Trigger was asserted.



Nonbuffered Edge Counting

The counter stops counting edges as soon as the Pause Trigger is asserted and resumes counting edges after the Pause Trigger is deasserted.



Physical Channels

This section contains information about physical channels for AO Series, bus-powered M Series, C Series, E Series, M Series, NI 6010, NI 6154, NI 6221 (37-pin), NI 623*X*, NI 653*X*, S Series, SCXI, SCC, TIO, and USB DAQ devices.

AO Series Physical Channels

Dev1 in physical channel names is the default device name for AO Series devices. You can change these names in MAX.

Analog Output

An AO Series device has between four and 32 analog output physical channels named Dev1/ao0 to Dev1/ao31.

With static AO devices, channels 0–15 are voltage output, and channels 16–31 are current output.

Digital Input and Output

All AO Series devices have eight lines of digital input and output named Dev1/port0/line0 through Dev1/port0/line7. These lines belong to a single port, and the physical channel Dev1/port0 refers to all eight lines at once.

Counter Input and Output

All AO Series devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are three primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	OUT Default
Dev1/ctr0	PFI 8	PFI 9	CTR 0 OUT
Dev1/ctr1	PFI 3	PFI 4	CTR 1 OUT

Refer to AO Series Signal Connections for Counters for more information.

C Series Physical Channels

In physical channel names, cDAQ1Mod1 is the default device name for a C Series device plugged into an NI cDAQ-9172 chassis, where cDAQ1 is the default chassis device name, and Mod1 refers to the slot number. You can change these names in MAX.

Analog Input

The NI 9225 has three physical channels named cDAQ1Mod1/ai0 to cDAQ1Mod1/ai2. The NI 9211, NI 9215, NI 9217, NI 9219, NI 9229, NI 9233, NI 9234, NI 9237, and NI 9239 have four analog input physical channels named cDAQ1Mod1/ai0 to cDAQ1Mod1/ai3. The NI 9201, NI 9203, NI 9221, NI 9235, and NI 9236 have eight analog input physical channels named cDAQ1Mod1/ai0 to cDAQ1Mod1/ai7.

The NI 9205 and NI 9206 have 32 analog input channels named cDAQ1Mod1/ai0 to cDAQ1Mod1/ai31. You can configure channels 0-7 and 16-23 as the positive channel of a differential pair. If *N* is this channel, channel *N* + 8 is the negative input of the pair. For instance, if you configure channel 1 in differential mode, the positive input is channel 1, and channel 9 is the negative input. Use only the physical channel name of the positive channel (not both) when creating a differential channel.

You can use channels from multiple analog input C Series devices in the same DAQmx task. Only one analog input task per chassis can run at a given time.

Strain and Wheatstone Bridge Measurements

The NI 9235, NI 9236, and NI 9237 support only the AI Strain Gage and AI Custom Voltage With Excitation channel types.

When using the NI 9219, NI 9235, NI 9236, or NI 9237 with an AI Custom Voltage With Excitation channel, you must set the **AI.Excit.UseForScaling** attribute/property to true. This attribute/property causes the channel to return ratiometric data: Vin/Vex. The NI 9219, NI 9235, NI 9236, and NI 9237 modules perform this division in hardware.

For the NI 9219, NI-DAQmx requires the **AI.Excit.Val** attribute/property to be set to 2.5 V for AI Strain Gage and AI Custom Voltage With Excitation channel types and to 500 µA for resistance and RTD measurements. The actual excitation voltage or current output by the NI 9219 varies with the sensor resistance or the load being measured.

With the NI 9219, NI 9235, NI 9236, and NI 9237, the **AI.Bridge.InitialVoltage** attribute/property refers to a voltage, not a ratio, so it should be set to the ratio Vin/Vex returned by the device multiplied by Vex.

The NI 9219 does not have quarter bridge completion circuitry, which

affects AI Strain Gage Quarter Bridge I channels and AI Custom Voltage With Excitation Quarter Bridge channels (but not AI Strain Gage Quarter Bridge II channels). With these channels, the NI 9219 performs a 2-wire resistance measurement on the active gage element, then NI-DAQmx uses software scaling to convert the resistance measurement into a bridge ratio. For these channels, the polynomial coefficients specified by the **AI.DevScalingCoeff** attribute/property convert unscaled data into Ohms, not into V/V. Likewise, the **AI.Rng.High/AI.Rng.Low** attributes/properties should be specified in units of Ohms, not V/V.

When the NI 9219 is in quarter bridge mode, you need to use the **AI.Bridge.NomResistance** attribute/property to control whether the channel uses the 120 Ω range or the 350 Ω range.

Analog Output

The NI 9263 and NI 9265 have has four analog output physical channels named cDAQ1Mod1/ao0 to cDAQ1Mod1/ao3. The NI 9264 has sixteen analog output physical channels named cDAQ1Mod1/ao0 to cDAQ1Mod1/ao15.

You can use channels from multiple analog output C Series devices in the same analog output task. If the task is hardware-timed, there is a limit of 16 channels per task, but if the task is software-timed, the number of channels is limited only by the number of devices. Only one hardware-timed analog output task per chassis can run at a given time.

When using the the NI 9263, NI 9264, or NI 9265, you can run only one type of timing at a time. You can have one software-timed task per channel or one hardware-timed task running on a device at one time, but you cannot have a combination of timing on that device. For instance, you can run up to four software-timed tasks on an the NI 9265 concurrently, but running one hardware-timed task with one software-timed task generates an error.

Digital Input and Output

The NI 9402, NI 9435, and 9481 have four lines of digital input and/or output named cDAQ1Mod1/port0/line0 to cDAQ1Mod1/port0/line3. The NI 9411 has six lines of digital input named cDAQ1Mod1/port0/line0 to cDAQ1Mod1/port0/line5. The NI 9401, NI 9421, NI 9422, NI 9423, NI 9472, NI 9474, and NI 9485 have eight lines of digital input and/or output named cDAQ1Mod1/port0/line0 to cDAQ1Mod1/port0/line7. These lines belong to a single port, and the physical channel cDAQ1Mod1/port0 refers to all four, six, or eight lines at once.

In the NI cDAQ-9172, C Series devices in slots 1, 2, 3, and 4 can perform both hardware-timed and static digital operations. You can use devices in slots 5 and 6 as static digital I/O lines or PFI lines, /cDAQ1Mod5/PFI0 to /cDAQ1Mod5/PFI7 and /cDAQ1Mod6/PFI0 to /cDAQ1Mod6/PFI7. Devices in slots 7 and 8 can be used only as static digital I/O lines. Hardware-timed digital input/output and PFI lines are only supported on correlated digital I/O modules. Refer to Digital I/O Considerations for C Series for more information.

Counter Input and Output

The NI cDAQ-9172 has two counter/timers that are used with a C Series digital I/O device in slot 5 or 6. These are referred to by the physical channel names cDAQ1Mod5/ctr0, cDAQ1Mod5/ctr1, cDAQ1Mod6/ctr0, and cDAQ1Mod6/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the device I/O connector but instead to circuits within the chassis. The NI cDAQ-9172 chassis also has a 4-bit frequency output generator, referred to as cDAQ1Mod5/freqout and cDAQ1Mod6/freqout.

Each counter has four primary terminals associated with it. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE, AUX, or GATE function and wire your signal to the default, you do not have to set the Input Terminal attribute/property.

Counter/timers are only supported on correlated digital I/O modules. Refer to Digital I/O Considerations for C Series for more information.

The following table shows the counter terminal defaults for 8-channel DIO/DI/DO C Series devices.

PFI Signal	Physical Channel Name
PFI 0	ctr0 src
PFI 1	ctr0 gate
PFI 2	ctr0 aux/freqout
PFI 3	ctr0 out
PFI 4	ctr1 src
PFI 5	ctr1 gate
PFI 6	ctr1 aux
PFI 7	ctr1 out

The following table shows the counter terminal defaults for 6-channel DIO/DI/DO C Series devices.

PFI Signal	Physical Channel Name
PFI 0	ctr0 src

PFI 1	ctr0 gate/freqout
PFI 2	ctr0 aux/ctr0 out
PFI 3	ctr1 src
PFI 4	ctr1 gate
PFI 5	ctr1 aux/ctr1 out

The following table shows the counter terminal defaults for 4-channel DIO/DI/DO C Series devices.

PFI Signal	**Physical Channel Name**
PFI 0	ctr0 src/ctr0 out
PFI 1	ctr0 gate/ctr1 aux/freqout
PFI 2	ctr0 aux/ctr1 gate
PFI 3	ctr1 src/ctr1 out

Refer to C Series Signal Connections for Counters for more information.

E Series Physical Channels

Dev1 in physical channel names is the default device name for E Series devices. You can change these names in MAX.

Analog Input

A 16-channel E Series device has physical channels ranging from Dev1/ai0 to Dev1/ai15. You can configure only channels 0 through 7 in differential mode. When you configure a channel in differential mode, the channel is the positive input and channel plus eight is the negative input. For instance, if you configure channel 1 in differential mode, the positive input is channel 1, and channel 9 is the negative input.

A 64-channel E Series device has physical channels ranging from Dev1/ai0 to Dev1/ai63. You can configure channels in banks of every other eight beginning with 0 through 7 as the positive channel of a differential pair (0-7, 16-23, 32-39, and 48-55). If *N* is this channel, channel *N* + 8 is the negative input of the pair.

Use only the physical channel name of the positive channel when creating a differential channel (not both).

Analog Output

An E Series device that supports analog output has two analog output physical channels named Dev1/ao0 and Dev1/ao1.

Digital Input and Output

All E Series devices except the NI 6025E have eight lines of digital input and output named Dev1/port0/line0 through Dev1/port0/line7. These lines belong to a single port, and the physical channel Dev1/port0 refers to all eight lines at once.

The NI 6025E has 32 lines of digital input and output with eight lines belonging to one of four ports. The names are of the form Dev1/port*P*/line0 through Dev1/port*P*/line7, where *P* ranges from 0 through 3. There are also four physical channel names that refer to all eight lines in a port at once of the form Dev1/portP, where *P* ranges from 0 through 3. There are two more physical channel names that refer to all the lines in multiple consecutive ports. They are both of the form Dev1/port*P*_*N*, where *P* is the port number of the lowest numbered port, and *N* is the total number of lines. All 32 lines at once can be configured as a single virtual channel with the physical channel name Dev1/port0_32. You can configure the two ports that can be handshaked as a single virtual channel by using the physical channel name Dev1/port1_16.

Counter Input and Output

All E Series devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are three primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	OUT Default
Dev1/ctr0	PFI 8	PFI 9	CTR 0 OUT
Dev/ctr1	PFI 3	PFI 4	CTR 1 OUT

Refer to E Series Signal Connections for Counters for more information.

M Series Physical Channels

In physical channel names, Dev1 is the default device name for M Series devices. You can change these names in MAX.

Analog Input

Depending on your M Series device, you can have from 16 to 80 analog input channels. A 16-channel M Series device has physical channels ranging from Dev1/ai0 to Dev1/ai15, a 32-channel device from Dev1/ai0 to Dev1/ai31, and so on. You can configure the first eight channels as the positive channel of a differential pair. If *N* is this channel, channel *N* + 8 is the negative input of the pair. For instance, if you configure channel 1 in differential mode, the positive input is channel 1, and channel 9 is the negative input. For devices with more than 16 AI channels, 16-23, 32-39, 48-55, and 64-71 are also positive channels of a differential pair.

Use only the physical channel name of the positive channel (not both) when creating a differential channel.

Analog Output

An M Series device that supports two analog outputs has analog output physical channels named Dev1/ao0 and Dev1/ao1.

An M Series device that supports four analog outputs has analog output physical channels named Dev1/ao0, Dev1/ao1, Dev1/ao2, and Dev1/ao3.

Digital Input and Output

All M Series devices have eight, 16, or 32 lines of digital input and output named Dev1/port0/line0 through Dev1/port0/line7, Dev1/port0/line0 through Dev1/port0/line15, or Dev1/port0/line0 through Dev1/port0/line31. These lines belong to a single port, and the physical channel Dev1/port0 refers to all eight or 32 lines at once. Port 0 can perform both hardware-timed and static digital operations.

M Series devices have two more ports, port 1 and port 2. Port 1 has eight digital I/O lines, Dev1/port1/line0 through Dev1/port1/line7. Port 2 has eight digital I/O lines, Dev1/port2/line0 through Dev1/port2/line7. Port 1 and port 2 can be used as static digital I/O lines or PFI lines, PFI 0..15. When any of PFI lines 0..15 is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	Physical Channel Name

PFI 0	Dev1/port1/line0
PFI 1	Dev1/port1/line1
PFI 2	Dev1/port1/line2
PFI 3	Dev1/port1/line3
PFI 4	Dev1/port1/line4
PFI 5	Dev1/port1/line5
PFI 6	Dev1/port1/line6
PFI 7	Dev1/port1/line7
PFI 8	Dev1/port2/line0
PFI 9	Dev1/port2/line1
PFI 10	Dev1/port2/line2
PFI 11	Dev1/port2/line3
PFI 12	Dev1/port2/line4
PFI 13	Dev1/port2/line5
PFI 14	Dev1/port2/line6
PFI 15	Dev1/port2/line7

Physical channel Dev1/port1 refers to all eight lines, Dev1/port1/line0:7, at once. Physical channel Dev1/port2 refers to all eight lines, Dev1/port2/line0:7, at once.

Counter Input and Output

All M Series devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are four primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE, AUX, or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 8	PFI 9	PFI 10	PFI 12

Dev1/ctr1	PFI 3	PFI 4	PFI 11	PFI 13

Refer to M Series Signal Connections for Counters for more information.

Bus-Powered M Series Physical Channels

In physical channel names, Dev1 is the default device name for M Series devices. You can change these names in MAX.

Analog Input

Depending on your M Series device, you can have from 16 to 32 analog input channels. A 16-channel M Series device has physical channels ranging from Dev1/ai0 to Dev1/ai15 and a 32-channel device from Dev1/ai0 to Dev1/ai31. You can configure the first eight channels as the positive channel of a differential pair. If *N* is this channel, channel *N* + 8 is the negative input of the pair. For instance, if you configure channel 1 in differential mode, the positive input is channel 1, and channel 9 is the negative input. For devices with more than 16 AI channels, 16-23, 32-39, 48-55, and 64-71 are also positive channels of a differential pair.

Use only the physical channel name of the positive channel (not both) when creating a differential channel.

Analog Output

An M Series device that supports two analog outputs has analog output physical channels named Dev1/ao0 and Dev1/ao1.

An M Series device that supports four analog outputs has analog output physical channels named Dev1/ao0, Dev1/ao1, Dev1/ao2, and Dev1/ao3.

Digital Input and Output

Most bus-powered M Series devices have two ports, port 0 and port 1. For devices with eight PFI lines, Port 0 has four digital input lines, Dev1/port0/line0 through Dev1/port0/line3, and port 1 has four digital output lines, Dev1/port1/line0 through Dev1/port1/line3. For devices with 16 PFI lines such as the NI 6218, Port 0 has eight digital input lines, Dev1/port0/line0 through Dev1/port0/line7, and port 1 has eight digital output lines, Dev1/port1/line0 through Dev1/port1/line7. You can use port 0 as static digital input lines or input PFI lines. You can use port 1 as static digital output lines or output PFI lines. When any of PFI lines 0..15 is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	Physical Channel Name
PFI 0	Dev1/port0/line0

PFI 1	Dev1/port0/line1
PFI 2	Dev1/port0/line2
PFI 3	Dev1/port0/line3
PFI 4	Dev1/port1/line0
PFI 5	Dev1/port1/line1
PFI 6	Dev1/port1/line2
PFI 7	Dev1/port1/line3
PFI 8	Dev1/port0/line4
PFI 9	Dev1/port0/line5
PFI 10	Dev1/port0/line6
PFI 11	Dev1/port0/line7
PFI 12	Dev1/port1/line4
PFI 13	Dev1/port1/line5
PFI 14	Dev1/port1/line6
PFI 15	Dev1/port1/line7

Physical channel Dev1/port0 refers to all four or eight lines, Dev1/port1/line0:3 or Dev1/port1/line0:7, at once. Physical channel Dev1/port1 refers to all four or eight lines, Dev1/port1/line0:3 or Dev1/port1/line0:7, at once.

The NI 6212 and NI 6216 have 16 lines of digital input and output named Dev1/port0/line0 through Dev1/port0/line15. These lines belong to a single port, and the physical channel Dev1/port0 refers to all 16 lines at once. Port 0 can perform static digital I/O operations only.

The NI 6212 and NI 6216 have two more ports, port 1 and port 2. Port 1 has eight digital I/O lines, Dev1/port1/line0 through Dev1/port1/line7. Port 2 has eight digital I/O lines, Dev1/port2/line0 through Dev1/port2/line7. Port 1 and port 2 can be used as static digital I/O lines or PFI lines, PFI 0..15. When any of PFI lines 0..15 is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	**Physical Channel Name**
PFI 0	Dev1/port1/line0
PFI 1	Dev1/port1/line1

PFI 2	Dev1/port1/line2
PFI 3	Dev1/port1/line3
PFI 4	Dev1/port1/line4
PFI 5	Dev1/port1/line5
PFI 6	Dev1/port1/line6
PFI 7	Dev1/port1/line7
PFI 8	Dev1/port2/line0
PFI 9	Dev1/port2/line1
PFI 10	Dev1/port2/line2
PFI 11	Dev1/port2/line3
PFI 12	Dev1/port2/line4
PFI 13	Dev1/port2/line5
PFI 14	Dev1/port2/line6
PFI 15	Dev1/port2/line7

Physical channel Dev1/port1 refers to all eight lines, Dev1/port1/line0:7, at once. Physical channel Dev1/port2 refers to all eight lines, Dev1/port2/line0:7, at once.

Counter Input and Output

All M Series devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are four primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE, AUX, or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

16-PFI Line Devices (NI 6218)

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 0	PFI 1	PFI 9	PFI 4
Dev1/ctr1	PFI 3	PFI 2	PFI 10	PFI 5

16-PFI Line Devices (NI 6212/6216)

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 8	PFI 9	PFI 10	PFI 12
Dev1/ctr1	PFI 3	PFI 4	PFI 11	PFI 13

8-PFI Line Devices (Such as the NI 6210/6211/6215)

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 0	PFI 1	PFI 0	PFI 4
Dev1/ctr1	PFI 3	PFI 2	PFI 3	PFI 5

Refer to Bus-Powered M Series Signal Connections for Counters for more information.

NI 6221 (37-Pin) Device Physical Channels

In physical channel names, Dev1 is the default device name for NI 6221 (37-pin) devices. You can change these names in MAX.

Analog Input

A 16-channel NI 6221 (37-pin) device has physical channels ranging from Dev1/ai0 to Dev1/ai15. You can configure only channels 0 through 7 in differential mode. When you configure a channel in differential mode, the channel is the positive input and channel plus eight is the negative input. For instance, if you configure channel 1 in differential mode, the positive input is channel 1, and channel 9 is the negative input.

Use only the physical channel name of the positive channel (not both) when creating a differential channel.

Analog Output

NI 6221 (37-pin) devices have two analog outputs corresponding to two analog output physical channels named Dev1/ao0 and Dev1/ao1.

Digital Input and Output

NI 6221 (37-pin) devices have two ports, port 0 and port 1. Port 0 has two digital I/O lines, Dev1/port0/line0 and Dev1/port0/line1. Port 1 has eight digital I/O lines, Dev1/port1/line0 through Dev1/port1/line7. Port 1 can be used as static digital I/O lines or input PFI lines, PFI 0..7. When any of PFI lines 0..7 is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	Physical Channel Name
PFI 0	Dev1/port1/line0
PFI 1	Dev1/port1/line1
PFI 2	Dev1/port1/line2
PFI 3	Dev1/port1/line3
PFI 4	Dev1/port1/line4
PFI 5	Dev1/port1/line5
PFI 6	Dev1/port1/line6
PFI 7	Dev1/port1/line7

Physical channel Dev1/port0 refers to both lines, Dev1/port0/line0:1, at

once. Physical channel Dev1/port1 refers to all eight lines, Dev1/port1/line0:7, at once.

Counter Input and Output

NI 6221 (37-pin) devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are four primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE, AUX, or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 0	PFI 1	PFI 2	PFI 6
Dev1/ctr1	PFI 3	PFI 4	PFI 5	PFI 7

In addition, the NI 6221 (37-pin) has one frequency generator. The output terminal of the frequency generator is FREQOUT. The default for FREQOUT is PFI 5. When using FREQOUT, you can continue to use both ctr0 and ctr1 to perform other operations.

NI 623*X* Physical Channels

In physical channel names, Dev1 is the default device name for NI 623*X* devices. You can change these names in MAX.

Analog Input

Refer to the following table for the physical channel naming conventions for each 623*X* device.

Device	Differential		RSE	
	+	-	+	-
NI 6230	Dev1/ai*n*	Dev1/ai(*n*+4)	Dev1/ai0..Dev1/ai*n*, where *n*=0..3	AI Gnd
NI 6232	Dev1/ai*n*	Dev1/ai(*n*+8)	Dev1/ai0..Dev1/ai*n*, where *n*=0..7	AI Gnd
NI 6233	Dev1/ai*n*	Dev1/ai(*n*+8)	Dev1/ai0..Dev1/ai*n*, where *n*=0..7	AI Gnd
NI 6236	Dev1/ai*n*+	Dev1/ai*n*-	Dev1/ai0..Dev1/ai*n*, where *n*=0..3	AI Gnd
NI 6238	Dev1/ai*n*+	Dev1/ai*n*-	Dev1/ai0..Dev1/ai*n*, where *n*=0..7	AI Gnd
NI 6239	Dev1/ai*n*+	Dev1/ai*n*-	Dev1/ai0..Dev1/ai*n*, where *n*=0..7	AI Gnd

Analog Output

NI 623*X* devices have *m* analog outputs corresponding to *m* analog output physical channels ranging from Dev1/ao0 to Dev1/ao(*m*-1). Refer to the hardware documentation for the number of analog outputs for your device.

Digital Input and Output

NI 623*X* devices have two ports, port 0 and port 1. Port 0 has six digital input lines, Dev1/port0/line0 through Dev1/port0/line5. Port 1 has four digital output lines, Dev1/port1/line0 through Dev1/port1/line3. You can use port 0 as static digital input lines or input PFI lines, PFI 0..5. You can use port 1 as static digital output lines or output PFI lines, PFI 6..9. When any of PFI lines 0..9 is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	Physical Channel Name	Direction
PFI 0	Dev1/port0/line0	Input
PFI 1	Dev1/port0/line1	Input
PFI 2	Dev1/port0/line2	Input

PFI 3	Dev1/port0/line3	Input
PFI 4	Dev1/port0/line4	Input
PFI 5	Dev1/port0/line5	Input
PFI 6	Dev1/port1/line0	Output
PFI 7	Dev1/port1/line1	Output
PFI 8	Dev1/port1/line2	Output
PFI 9	Dev1/port1/line3	Output

Physical channel Dev1/port0 refers to all six input lines, Dev1/port0/line0:5, at once. Physical channel Dev1/port1 refers to all four output lines, Dev1/port1/line0:3, at once.

Tristating Digital Output Channels (NI 6230/6236)

NI 6230/6236 devices support tristating for port 1, the four digital output lines. The power-on state default is to tristate port 1. Tristating is supported only for the entire port at a time, not on a per-line basis. For instance, port 1 remains tristated as long as no lines on port 1 are toggled to generate a value. After a line on port 1 is toggled, all lines on the port are driven to logic high or logic low depending on what you choose. The default is logic low.

Counter Input and Output

NI 623*X* devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are four primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE, AUX, or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 0	PFI 1	PFI 2	PFI 6
Dev1/ctr1	PFI 3	PFI 4	PFI 5	PFI 7

In addition, the NI 623*X* has one frequency generator. The output

terminal of the frequency generator is FREQOUT. The default for FREQOUT is PFI 8. When using FREQOUT, you can continue to use both ctr0 and ctr1 to perform other operations.

The NI 623*X* uses the same default terminals for common counter applications as other 37-Pin DSUB devices. Refer to 37-Pin DSUB Signal Connections for Counters for default terminals for NI 623*X* devices.

NI 6010 Physical Channels

In physical channel names, Dev1 is the default device name for NI 6010 devices. You can change these names in MAX.

Analog Input

A 16-channel NI 6010 device has physical channels ranging from Dev1/ai0 to Dev1/ai15. You can configure only channels 0 through 7 in differential mode. When you configure a channel in differential mode, the channel is the positive input and channel plus eight is the negative input. For instance, if you configure channel 1 in differential mode, the positive input is channel 1, and channel 9 is the negative input.

Use only the physical channel name of the positive channel (not both) when creating a differential channel.

Analog Output

NI 6010 devices have two analog outputs corresponding to two analog output physical channels named Dev1/ao0 and Dev1/ao1.

Digital Input and Output

NI 6010 devices have two ports, port 0 and port 1. Port 0 has six digital input lines, Dev1/port0/line0 through Dev1/port0/line5. Port 1 has four digital output lines, Dev1/port1/line0 through Dev1/port1/line3. Port 0 can be used as static digital input lines or input PFI lines, PFI 0..5. Port 1 can be used as static digital output lines or output PFI lines, PFI 6..9. When any of PFI lines 0..9 is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	Physical Channel Name
PFI 0	Dev1/port0/line0
PFI 1	Dev1/port0/line1
PFI 2	Dev1/port0/line2
PFI 3	Dev1/port0/line3
PFI 4	Dev1/port0/line4
PFI 5	Dev1/port0/line5
PFI 6	Dev1/port1/line0
PFI 7	Dev1/port1/line1

PFI 8	Dev1/port1/line2
PFI 9	Dev1/port1/line3

Physical channel Dev1/port0 refers to all six lines, Dev1/port0/line0:5, at once. Physical channel Dev1/port1 refers to all four lines, Dev1/port1/line0:3, at once.

Counter Input and Output

NI 6010 devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are four primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE, AUX, or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 0	PFI 1	PFI 2	PFI 6
Dev1/ctr1	PFI 3	PFI 4	PFI 5	PFI 7

Refer to 37-Pin DSUB Signal Connections for Counters for more information.

NI 6154 Physical Channels

In physical channel names, Dev1 is the default device name for NI 6154 devices. You can change these names in MAX.

Analog Input

A 4-channel NI 6154 device has physical channels ranging from Dev1/ai0 to Dev1/ai3.

Analog Output

NI 6154 devices have four analog outputs corresponding to four analog output physical channels ranging from Dev1/ao0 to Dev1/ao3.

Digital Input and Output

NI 6154 devices have two ports, port 0 and port 1. Port 0 has six digital input lines, Dev1/port0/line0 through Dev1/port0/line5. Port 1 has four digital output lines, Dev1/port1/line0 through Dev1/port1/line3. You can use port 0 as static digital input lines or input PFI lines, PFI 0..5. You can use port 1 as static digital output lines or output PFI lines, PFI 6..9. When any of PFI lines 0..9 is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	Physical Channel Name	Direction
PFI 0	Dev1/port0/line0	Input
PFI 1	Dev1/port0/line1	Input
PFI 2	Dev1/port0/line2	Input
PFI 3	Dev1/port0/line3	Input
PFI 4	Dev1/port0/line4	Input
PFI 5	Dev1/port0/line5	Input
PFI 6	Dev1/port1/line0	Output
PFI 7	Dev1/port1/line1	Output
PFI 8	Dev1/port1/line2	Output
PFI 9	Dev1/port1/line3	Output

Physical channel Dev1/port0 refers to all six input lines, Dev1/port0/line0:5, at once. Physical channel Dev1/port1 refers to all four output lines, Dev1/port1/line0:3, at once.

Counter Input and Output

NI 6154 devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are four primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE, AUX, or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 0	PFI 1	PFI 2	PFI 6
Dev1/ctr1	PFI 3	PFI 4	PFI 5	PFI 7

In addition, the NI 6154 has one frequency generator. The output terminal of the frequency generator is FREQOUT. The default for FREQOUT is PFI 8. When using FREQOUT, you can continue to use both ctr0 and ctr1 to perform other operations.

The NI 6154 uses the same default terminals for common counter applications as other 37-Pin DSUB devices. Refer to 37-Pin DSUB Signal Connections for Counters for default terminals for NI 6154 devices.

NI 653X Device Physical Channels

Digital Input and Output

All NI 653*X* devices have 32 individually configurable lines of digital input and output that are grouped into four 8-bit ports.

Port	NI-DAQmx Physical Channel Name (Lines)	NI-DAQmx Physical Channel Name (Ports)[1]
Port 0	Dev1/port0/line0 — Dev1/port0/line7	Dev1/port0
Port 1	Dev1/port1/line0 — Dev1/port1/line7	Dev1/port1
Port 2	Dev1/port2/line0 — Dev1/port2/line7	Dev1/port2
Port 3	Dev1/port3/line0 — Dev1/port3/line7	Dev1/port3
[1]This physical channel name refers to all eight lines in a port at once.		

For Ports 0 through 3, you can configure a port width of eight, 16, or 32 bits. To configure a 32-bit port, use the physical channel name Dev1/port0_32. To configure a 16-bit port, use channel names that refer to all the lines in multiple consecutive ports: Dev1/portP_N, where *P* is the port number of the lowest numbered port, and *N* is the total number of lines. For instance, combining Port 2 and 3 into a 16-bit port, you would specify Dev1/port2_16 as the physical channel.

NI 653*X* devices also have eight fixed-direction lines, grouped into two ports, that use PFI lines. Port 4 is used for input operations; Port 5 is for output.

Port 4 and port 5 can be used as static digital I/O lines or PFI lines. When any of these PFI lines is used as a digital I/O signal, it uses the physical channel name shown in the following table.

PFI Signal	Physical Channel Name
PFI0	Dev1/port4/line0
PFI1	Dev1/port4/line1
PFI2	Dev1/port4/line2

PFI3	Dev1/port4/line3
PFI4	Dev1/port5/line2
PFI5	Dev1/port5/line3
PFI6	Dev1/port5/line0
PFI7	Dev1/port5/line1

S Series Physical Channels

Dev1 in physical channel names is the default device name for S Series devices. You can change these names in MAX.

Analog Input

An S Series device has between two and eight analog input physical channels named Dev1/ai0 to Dev1/ai7.

Analog Output

An S Series device that supports analog output has two analog output physical channels named Dev1/ao0 and Dev1/ao1.

Digital Input and Output

All S Series devices have eight lines of digital input and output named Dev1/port0/line0 through Dev1/port0/line7. These lines belong to a single port, and the physical channel Dev1/port0 refers to all eight lines at once.

Counter Input and Output

All S Series devices have two counter/timers referred to by the physical channel names Dev1/ctr0 and Dev1/ctr1. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device. There are three primary terminals associated with each counter. These are the terminals used as the SOURCE, GATE and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signal provides the SOURCE or GATE function and wire your signal to the default, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	OUT Default
Dev1/ctr0	PFI 8	PFI 9	CTR 0 OUT
Dev/ctr1	PFI 3	PFI 4	CTR 1 OUT

Refer to S Series Signal Connections for Counters for more information.

SCXI and SCC Physical Channels

SC1Mod1 is the default device name for an SCXI module, where SC1 is the default chassis ID, and Mod1 refers to the slot number. These names can be changed in MAX.

Analog Input

An SCXI module usually has eight or 32 analog input channels; refer to your device documentation to be sure. These physical channel names are of the form SC1Mod<*slot#*>/ai0 to SC1Mod<*slot#*>/ai*N*, where <*slot#*> is the chassis slot number of the module, and *N* equals the number of analog input channels on the module minus one. For example, SCI1Mod1/ai31 is the highest numbered physical channel for a module with 32 analog input channels.

An SCC module has either one or two physical channels named SCC1Mod<*J connector#*>ai*N*, where <*J connector#*> is the number of the J connector where the SCC module resides, and *N* is the channel number. SCC1 is the SCC connector block ID (for example, SCC1Mod1/ai0).

NI PXI-4224 Only—You cannot scan channel ai7 and the CJC channel simultaneously in a task, since the CJC channel is multiplexed to channel 7. However, when you make a thermocouple measurement on ai0:7 with internal CJC, NI-DAQmx automatically reads the CJC channel at the beginning of the measurement and then scans the rest of the channels correctly.

Analog Output

An SCXI module has some number of output channels for voltage or current. These physical channel names are of the form SC1Mod<*slot#*>/ao0 to SC1Mod<*slot#*>/ao*N*, where <*slot#*> is the chassis slot number of the module, and *N* equals the number of analog output channels on the module minus one. For example, SC1Mod1/ao5 is the highest numbered physical channel on a module with six channels.

Digital Input and Output

An SCXI digital module has eight, 16, or 32 lines named SC1Mod<*slot#*>/port0/line0 through SC1Mod<*slot#*>/port0/line*N*, where <*slot#*> is the chassis slot number of the module, and *N* is the number of digital lines minus one. For example, SC1Mod1/port0/line31 is the highest numbered line for a module with 32 lines. These lines belong to a single port and the physical channel named SC1Mod<*slot#*>/port0 refers to all the lines at once.

An SCC module has either one digital input line or one digital output line with names of the form SCC1Mod<*J connector#*>di*N* or SCC1Mod<*J connector#*>di*N*, where <*J connector#*> is the number of the J connector where the SCC module resides, and *N* is the channel number. SCC1 is the SCC connector block ID (for example, SCC1Mod1/di0).

SensorDAQ Physical Channels

SensorDAQ has analog input, analog output, digital I/O, and counter channels. It also has three analog sensor channels (labeled Ch. 1, Ch. 2, and Ch. 3 on the device) and one digital sensor channel (labeled DIG on the device) for use with Vernier sensors. Please contact Vernier for additional information on the analog and digital sensor channels.

Analog Input

SensorDAQ has two AI physical channels, Dev1/ai0 and Dev1/ai1. When you configure a channel in differential mode, Dev1/ai0 is the positive input and Dev1/ai1 is the negative input.

Analog Output

SensorDAQ has two AO physical channels, Dev1/ao0 and Dev1/ao1.

Digital Input/Output

SensorDAQ has four digital I/O physical channels.

Counter Input and Output

There is one counter channel referred to by the physical channel name Dev1/ctr0. pfi0 is the terminal used for this physical channel.

TIO Physical Channels

Counter Input and Output

TIO devices have up to eight counter/timers referred to by the physical channel names Dev1/ctr0 to Dev1/ctr7. Unlike the other I/O types, these physical channel names do not refer to terminals on the I/O connector but instead to circuits within the device.

There are four primary terminals associated with each TIO counter. These are the terminals used as the SOURCE, GATE, AUX, and OUT functions. NI-DAQmx has default values for these terminals. For counter input tasks, if you know whether your signals provide the SOURCE, GATE, or AUX functions and wire your signal to the default input, you do not have to set the **Input Terminal** attribute/property.

Counter	SOURCE Default	GATE Default	AUX Default	OUT Default
Dev1/ctr0	PFI 39	PFI 38	PFI 37	PFI 36
Dev1/ctr1	PFI 35	PFI 34	PFI 33	PFI 32
Dev1/ctr2	PFI 31	PFI 30	PFI 29	PFI 28
Dev1/ctr3	PFI 27	PFI 26	PFI 25	PFI 24
Dev1/ctr4	PFI 23	PFI 22	PFI 21	PFI 20
Dev1/ctr5	PFI 19	PFI 18	PFI 17	PFI 16
Dev1/ctr6	PFI 15	PFI 14	PFI 13	PFI 12
Dev1/ctr7	PFI 11	PFI 10	PFI 9	PFI 8

Note The NI 6601 has only four counters (ctr0–ctr3). The entries in the previous table for cntr4, cntr5, cntr6, and cntr7 do not apply for that device.

USB DAQ Physical Channels

`Dev1` in physical channel names is the default device name for USB Series DAQ devices. You can change these names in MAX.

Analog Input

The NI USB-9211, NI USB-9211A, NI USB-9215, NI USB-9215A, NI USB-9219, NI USB-9229, NI USB-9233, and NI USB-9239 have four analog input physical channels named `Dev1/ai0` to `Dev1/ai3`. The NI USB-9201 and NI USB-9221 have eight analog input physical channels named `Dev1/ai0` to `Dev1/ai7`.

Strain and Wheatstone Bridge Measurements

The NI USB-9237 supports only the AI Strain Gage and AI Custom Voltage With Excitation channel types.

When using the NI USB-9219 or NI USB-9237 with an AI Custom Voltage With Excitation channel, you must set the **AI.Excit.UseForScaling** attribute/property to true. This attribute/property causes the channel to return ratiometric data: Vin/Vex. The NI USB-9219 and NI USB-9237 modules perform this division in hardware.

For the NI USB-9219, NI-DAQmx requires the **AI.Excit.Val** attribute/property to be set to 2.5 V for AI Strain Gage and AI Custom Voltage With Excitation channel types and to 500 µA for resistance and RTD measurements. The actual excitation voltage or current output by the NI USB-9219 varies with the sensor resistance or the load being measured.

With the NI USB-9219 and NI USB-9237, the **AI.Bridge.InitialVoltage** attribute/property refers to a voltage, not a ratio, so it should be set to the ratio Vin/Vex returned by the device multiplied by Vex.

The NI USB-9219 does not have quarter bridge completion circuitry, which affects AI Strain Gage Quarter Bridge I channels and AI Custom Voltage With Excitation Quarter Bridge channels (but not AI Strain Gage Quarter Bridge II channels). With these channels, the NI USB-9219 performs a 2-wire resistance measurement on the active gage element, then NI-DAQmx uses software scaling to convert the resistance measurement into a bridge ratio. For these channels, the polynomial coefficients specified by the **AI.DevScalingCoeff** attribute/property convert unscaled data into Ohms, not into V/V. Likewise, the **AI.Rng.High/AI.Rng.Low** attributes/properties should be specified in units of Ohms, not V/V.

When the NI USB-9219 is in quarter bridge mode, you need to use the

AI.Bridge.NomResistance attribute/property to control whether the channel uses the 120 Ω range or the 350 Ω range.

Analog Output

The NI USB-9263 has four analog output physical channels named Dev1/ao0 to Dev1/ao3.

When using the the NI USB-9263, you can run only one type of timing at a time. You can have one software-timed task per channel or one hardware-timed task running on a device at one time, but you cannot have a combination of timing on that device. For instance, you can run up to four software-timed tasks on an the NI USB-9263 concurrently, but running one hardware-timed task with one software-timed task generates an error. Additionally, the NI USB-9263 can run only one hardware-timed analog output task per device at a given time.

Internal Channels

This section contains information about internal channels for C Series, DSA, E Series, M Series, NI 6010, NI PXI-42*XX*, S Series, SCXI, and USB DAQ devices.

Internal Channels for C Series Devices

The following table lists the internal channels for the NI 9211 device.

Internal Channel Name	Description
_aignd_vs_aignd	A differential terminal with the positive and negative terminals both connected to the ground reference for analog input.
_calref_vs_aignd	A differential terminal with the positive terminal connected to the internal calibration reference voltage and the negative terminal connected to the ground reference for analog input.
_cjtemp	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input. This channel is used for cold-junction compensation.

The following table lists the internal channels for the NI 9205 and NI 9206 devices.

Internal Channel Name	Description
_aignd_vs_aignd	A differential terminal with the positive and negative terminals both connected to the ground reference for analog input.
_calref_vs_aignd	A differential terminal with the positive terminal connected to the internal calibration reference voltage and the negative terminal connected to the ground reference for analog input.
_aignd_vs_aisense	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected to physical channel AI SENSE.
_calSrcHi_vs_aignd	A differential terminal with the positive terminal connected to the calibration PWM

	and the negative terminal connected to the ground reference for analog input.
_calref_vs_calSrcHi	A differential terminal with the positive terminal connected to the internal calibration reference voltage and the negative terminal connected to the calibration PWM.
_calSrcHi_vs_calSrcHi	A differential terminal with the positive and negative terminals connected to the calibration PWM.
_aignd_vs_calSrcHi	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected to the calibration PWM.
_calSrcMid_vs_aignd	A differential terminal with the positive terminal connected to the calibration PWM and the negative terminal connected to the ground reference for analog input. _calSrcMid is the divided down version of _calSrcHi.
_boardTempSensor_vs_aignd	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input.
_ai0_vs_calSrcHi	A differential terminal with the positive terminal connected to physical channel ai0 and the negative terminal connected to the calibration PWM.
_ai8_vs_calSrcHi	A differential terminal with the positive terminal connected to physical channel ai8 and the negative terminal connected to the calibration PWM.

The following table lists the internal channels for the NI cDAQ-9172 chassis.

Internal	

Channel Name	Description
_ctr0	This physical channel name does not refer to a terminal on the I/O connector but instead to a circuit within the device. You must set the **Input Terminal** or **Output Terminal** attributes/properties that are appropriate for the measurement/generation being performed.
_ctr1	This physical channel name does not refer to a terminal on the I/O connector but instead to a circuit within the device. You must set the **Input Terminal** or **Output Terminal** attributes/properties that are appropriate for the measurement/generation being performed.
_freqout	This physical channel name does not refer to a terminal on the I/O connector but instead to a circuit within the device. You must set the **Output Terminal** attribute/property that is appropriate for the generation being performed.

The following table lists the internal channels for the NI 9219 device.

Internal Channel Name	Description
_cjtemp0	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input. This channel is used for cold-junction compensation for analog input channel 0.
_cjtemp1	This channel is used for cold-junction compensation for analog input channel 1.
_cjtemp2	This channel is used for cold-junction compensation for analog input channel 2.
_cjtemp3	This channel is used for cold-junction compensation for analog input channel 3.

Internal Channels for DSA Devices

On a DSA device, you can either acquire a signal present on the I/O connector, or you can acquire a signal that is generated from the internal calibration multiplexer. The channels available on this multiplexer are typically used for calibration purposes, but you can also sample them as you would a physical signal present on the I/O connector. To read from one of these internal channels, you must use one of the device's AI physical channels (for instance, Dev1/ai0) when creating the channel to select which ADC to use, then set the appropriate string value on the **Input Source** channel attribute/property.

The following table lists internal channels for DSA devices.

Internal Channel Name	Supported Devices	Description
_external_channel	All DSA devices	The source of the AI channel is the device input connector, or an accessory connected to the device connector.
_5Vref_vs_aignd	446*X*, 447*X*	The source of the AI channel is the onboard reference signal (for example, +5V).
_ao0_vs_ao0neg*	4461 only	The source of the AI channel is the onboard analog output channel 0.
_ao1_vs_ao1neg*	4461 only	The source of the AI channel is the onboard analog output channel 1.
_aignd_vs_aignd	446*X*, 447*X*, 449*X*	The source of the AI channel is the onboard ground signal.
_ref_sqwv_vs_aignd	449*X*	The source of the AI channel is the onboard reference square wave signal.
* The AO internal channels are valid only for devices with AO physical channels.		

For all DSA devices, only one internal channel can be read at a time, although the same internal channel can be read on multiple physical channels (with additional restrictions for NI 447*X* devices). For example, you cannot simultaneously read the internal 5 V reference on one

physical channel and the analog ground on another physical channel.

The NI 447*X* AI physical channels are grouped into pairs, for instance {ai0, ai1}, {ai2, ai3}, and so on. NI 447*X* devices cannot read an internal channel on more than one physical channel group, and when reading an internal channel, both physical channels in the group are connected to the internal channel source. For example, if the **Input Source** for channel ai0 is set to 5Vref_vs_aignd and the **Input Source** for channel ai1 is left at the default value of _external_channel, ai1 still reads the internal channel 5Vref_vs_aignd since ai0 and ai1 are in the same physical channel group.

Internal Channels for E Series Devices

The following table is a list of internal physical channels for all E Series devices. Different E Series devices have different subsets of channels. These channels are typically for self-calibration, and you can sample them as you would a physical channel present on the I/O connector.

Internal Channel Name	Description
_aognd_vs_aognd	A differential terminal with the positive and negative terminals both connected to the ground reference for analog output.
_aognd_vs_aignd	A differential terminal with the positive terminal connected to the ground reference for analog output and the negative terminal connected to the ground reference for analog input.
_ao0_vs_aognd	A differential terminal with the positive terminal connected to physical channel ao0 and the negative terminal connected to the ground reference for analog output.
_ao1_vs_aognd	A differential terminal with the positive terminal connected to physical channel ao1 and the negative terminal connected to the ground reference for analog output.
_calref_vs_calref	A differential terminal with the positive terminal connected to the onboard 5 V reference and the negative terminal connected to the onboard 5 V reference.
_calref_vs_aignd	A differential terminal with the positive terminal connected to the onboard 5 V reference and the negative terminal connected to the ground reference for analog input.
_ao0_vs_calref	A differential terminal with the positive terminal connected to physical channel ao0 and the negative terminal connected to the onboard 5 V reference.

_ao1_vs_calref	A differential terminal with the positive terminal connected to physical channel ao1 and the negative terminal connected to the onboard 5 V reference.
_ao1_vs_ao0	A differential terminal with the positive terminal connected to physical channel ao1 and the negative terminal connected to physical channel ao0.
_boardTempSensor_vs_aignd	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input.
_aignd_vs_aignd	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected the ground reference for analog input.
_caldac_vs_aignd	A differential terminal with the positive terminal connected to the onboard calibration DAC and the negative terminal connected to the ground reference for analog input.
_caldac_vs_calref	A differential terminal with the positive terminal connected to the onboard calibration DAC and the negative terminal connected to the onboard 5 V reference.
_PXI_SCXIbackplane_vs_aignd	A terminal where a signal being conditioned by a SCXI module is measured across the PXI/SCXI backplane and not the I/O connector. Reading from this channel is valid only on PXI devices inserted in the rightmost PXI slot of a PXI/SCXI combination chassis.

Internal Channels for M Series and NI 6010 Devices

The following table describes all M Series and NI 6010 internal channels.

Internal Channel Name	Description
_aignd_vs_aignd	A differential terminal with the positive and negative terminals both connected to the ground reference for analog input.
_ao0_vs_aognd	A differential terminal with the positive terminal connected to physical channel ao0 and the negative terminal connected to the ground reference for analog output.
_ao1_vs_aognd	A differential terminal with the positive terminal connected to physical channel ao1 and the negative terminal connected to the ground reference for analog output.
_ao2_vs_aognd	For M Series devices only—A differential terminal with the positive terminal connected to physical channel ao2 and the negative terminal connected to the ground reference for analog output.
_ao3_vs_aognd	For M Series devices only—A differential terminal with the positive terminal connected to physical channel ao3 and the negative terminal connected to the ground reference for analog output.
_calref_vs_aignd	A differential terminal with the positive terminal connected to the internal calibration reference voltage and the negative terminal connected to the ground reference for analog input.
_aignd_vs_aisense	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected to physical channel AI SENSE.

_aignd_vs_aisense2	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected to physical channel AI SENSE2.
_calSrcHi_vs_aignd	A differential terminal with the positive terminal connected to the calibration PWM and the negative terminal connected to the ground reference for analog input.
_calref_vs_calSrcHi	A differential terminal with the positive terminal connected to the internal calibration reference voltage and the negative terminal connected to the calibration PWM.
_calSrcHi_vs_calSrcHi	A differential terminal with the positive and negative terminals connected to the calibration PWM.
_aignd_vs_calSrcHi	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected to the calibration PWM.
_calSrcMid_vs_aignd	A differential terminal with the positive terminal connected to the calibration PWM and the negative terminal connected to the ground reference for analog input. _calSrcMid is the divided down version of _calSrcHi.
_calSrcLo_vs_aignd	A differential terminal with the positive terminal connected to the calibration PWM and the negative terminal connected to the ground reference for analog input. _calSrcLo is the divided down version of _calSrcHi.
_ai0_vs_calSrcHi	A differential terminal with the positive

	terminal connected to physical channel ai0 and the negative terminal connected to the calibration PWM.
_ai8_vs_calSrcHi	A differential terminal with the positive terminal connected to physical channel ai8 and the negative terminal connected to the calibration PWM.
_boardTempSensor_vs_aignd	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input.
_PXI_SCXIbackplane_vs_aignd	A terminal where a signal being conditioned by a SCXI module is measured across the PXI/SCXI backplane and not the I/O connector. Reading from this channel is valid only on PXI devices inserted in the rightmost PXI slot of a PXI/SCXI combination chassis.

Internal Channels for the NI PXI-42*XX*

The following table is a list of internal physical channels for the PXI-42*XX* devices. The subset of channels present on your device depends on the specific E Series device being used. These channels are typically for self-calibration, and you can sample them as you would a physical channel present on the I/O connector.

Internal Channel Name	Description
_cjTemp	A cold-junction compensation channel for measuring the temperature at the I/O connector when making thermocouple measurements.
_aignd_vs_aignd	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected the ground reference for analog input.
_caldac_vs_aignd	A differential terminal with the positive terminal connected to the onboard calibration DAC and the negative terminal connected to the ground reference for analog input.
_calref_vs_aignd	A differential terminal with the positive terminal connected to the onboard 5 V reference and the negative terminal connected to the ground reference for analog input.
_boardTempSensor_vs_aignd	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input.
PXI_SCXIbackplane_vs_aignd	A terminal where a signal being conditioned by a SCXI module is measured across the PXI/SCXI backplane and not the I/O connector. Reading from this

	channel is valid only on PXI devices inserted in the rightmost PXI slot of a PXI/SCXI combination chassis.
_extcal_vs_aignd	A differential terminal with the positive terminal connected to the external calibration input and the negative terminal connected to the ground reference for analog input.
_pod_calrefPos_vs_aignd	A differential terminal with the positive terminal connected to the internal positive reference voltage and the negative terminal connected to the ground reference for analog input.
_pod_calrefNeg_vs_aignd	A differential terminal with the positive terminal connected to the internal negative reference voltage and the negative terminal connected to the ground reference for analog input.

Internal Channels for S Series Devices

On an S Series device, you can either acquire a signal present on the I/O connector, or you can acquire a signal that is being generated from the internal calibration multiplexer. The channels available on this multiplexer are typically used for calibration purposes, but you can also sample them as you would a physical signal present on the I/O connector. To read from one of these internal channels, you must use one of the device's AI physical channels (Dev1/ai0 through Dev1/ai7) when creating the virtual channel. The physical channel specifies the ADC for the internal channel. You can then set the appropriate string value on the **Input Source** channel attribute/property.

Note All S Series devices must have the same **Input Source** setting on all channels. The NI PCI-6110 and NI PCI-6111 devices cannot acquire from more than one ADC at a time when using an internal channel.

NI PCI-6110, NI PCI-6111, NI 6115, NI 6120 Internal Channels

- _external_channel
- _aognd_vs_aognd
- _aognd_vs_aignd
- _ao0_vs_aognd
- _ao1_vs_aognd
- _calref_vs_calref
- _calref_vs_aignd
- _ao0_vs_calref
- _ao1_vs_calref

NI PCI-6143 Internal Channels

- _external_channel
- _aignd_vs_aignd
- _calref_vs_aignd
- _calSrcHi_vs_aignd
- _calref_vs_calSrcHi
- _calSrcHi_vs_calSrcHi
- _aignd_vs_calSrcHi

NI PXI-6132/6133 Internal Channels

- _external_channel
- _aignd_vs_aignd
- _calref_vs_aignd
- _calSrcMid_vs_aignd
- _calSrcHi_vs_aignd
- _calref_vs_calSrcHi
- _calSrcMid_vs_calSrcHi
- _calSrcHi_vs_calSrcHi
- _aignd_vs_calSrcHi

NI PCI-6154 Internal Channels

- _external_channel
- _aignd_vs_aignd
- _calref_vs_aignd
- _calSrcHi_vs_aignd
- _aignd_vs_calSrcHi
- _calref_vs_calSrcHi
- _calSrcHi_vs_calSrcHi
- _aox_vs_aognd

The following table describes all S Series internal channels.

Internal Channel Name	Description
_external_channel	The differential terminal on the I/O connector that is typically used for acquiring data.
_aignd_vs_aignd	A differential terminal with the positive and negative terminals both connected to the ground reference for analog input.
_aognd_vs_aognd	A differential terminal with the positive and negative terminals both connected to the ground reference for analog output.
_aognd_vs_aignd	A differential terminal with the positive terminal connected to the ground reference for analog output and the negative terminal connected to the ground reference for analog input.
_ao*x*_vs_aognd	A differential terminal with the positive terminal connected to the analog output physical channel, such as ao0, and the negative terminal connected to the ground reference for analog output.
_ao1_vs_aognd	A differential terminal with the positive terminal connected to physical channel ao1 and the negative terminal connected to the ground reference for analog output.
_calref_vs_calref	A differential terminal with the positive terminal

	connected to the internal calibration reference voltage and the negative terminal connected to the internal calibration reference voltage.
_calref_vs_aignd	A differential terminal with the positive terminal connected to the internal calibration reference voltage and the negative terminal connected to the ground reference for analog input.
_ao0_vs_calref	A differential terminal with the positive terminal connected to physical channel ao0 and the negative terminal connected to the internal calibration reference voltage.
_ao1_vs_calref	A differential terminal with the positive terminal connected to physical channel ao1 and the negative terminal connected to the internal calibration reference voltage.
_calSrcHi_vs_aignd	A differential terminal with the positive terminal connected to the calibration PWM and the negative terminal connected to the ground reference for analog input.
_calref_vs_calSrcHi	A differential terminal with the positive terminal connected to the internal calibration reference voltage and the negative terminal connected to the calibration PWM.
_calSrcHi_vs_calSrcHi	A differential terminal with the positive and negative terminals connected to the calibration PWM.
_aignd_vs_calSrcHi	A differential terminal with the positive terminal connected to the ground reference for analog input and the negative terminal connected to the calibration PWM.
_calSrcMid_vs_aignd	A differential terminal with the positive terminal connected to the calibration PWM and the negative terminal connected to the ground reference for analog input. _calSrcMid is the divided down version of _calSrcHi.
_calSrcMid_vs_calSrcHi	A differential terminal with the positive and

	negative terminals connected to the calibration PWM. _calSrcMid is the divided down version of _calSrcHi.

SCXI Internal Channels

Some SCXI modules also have internal channels. These are physical channels that are not accessible from an I/O connector. To measure the signals present at these internal physical channels, use them to create virtual channels. The SCXI-1100, SCXI-1102, SCXI-1120, SCXI-1121, SCXI-1122, and SCXI-1125 modules have an internal physical channel called _cjTemp channel. It is the cold-junction compensation channel for measuring the temperature at the connector for thermocouples.

The SCXI-1112 has internal channels _cjTemp0, _cjTemp1, _cjTemp2, _cjTemp3, _cjTemp4, _cjTemp5, _cjTemp6, and _cjTemp7. These are the cold-junction compensation channels for each analog input channel on the SCXI-1112.

The SCXI-1520 has eight pairs of internal channels _pPos0 and _pNeg0 through _pPos7 and _pNeg7. These channels read back the excitation on the corresponding analog input channel. The pPos half of the pair is the positive side of the excitation, and the pNeg half is the negative side of the excitation. The real excitation value is the pPos value minus the pNeg value.

The SCXI-1521/B has 24 voltage excitation internal channels _Vex0 through _Vex23. These channels read back the excitation on the corresponding analog input channel. In addition, the SCXI-1521/B has 24 pairs of internal channels _IexPos0 and _IexNeg0 through _IexPos23 and _IexNeg23. The current through a sensor connected to channel *X* is the _IexPos*X* value minus the _IexNeg*X* value.

Internal Channels for USB DAQ Devices

The following table lists the internal channel for the USB-9211 device.

Internal Channel Name	Description
_cjtemp	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input. This channel is used for cold-junction compensation.

The following table lists the internal channels for the USB-9219 device.

Internal Channel Name	Description
_cjtemp0	A differential terminal with the positive terminal connected to the onboard temperature sensor and the negative terminal connected to the ground reference for analog input. This channel is used for cold-junction compensation for analog input channel 0.
_cjtemp1	This channel is used for cold-junction compensation for analog input channel 1.
_cjtemp2	This channel is used for cold-junction compensation for analog input channel 2.
_cjtemp3	This channel is used for cold-junction compensation for analog input channel 3.

Default Input/Output Terminal Configurations

If you do not explicitly specify the input or output terminal configuration when you create a channel, NI-DAQmx automatically determines the default terminal configuration at run time. The following table lists the default terminal configurations for devices.

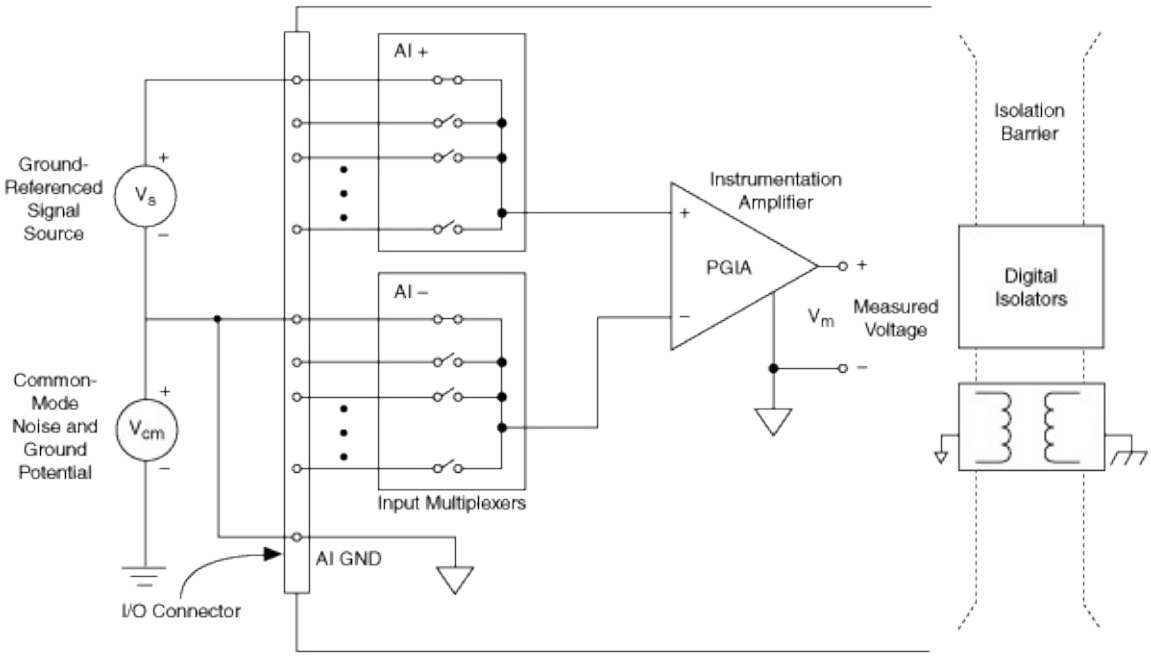
Device	Default Input Terminal Configuration	Default Output Terminal Configuration
AO Series	N/A	Referenced single-ended
DSA	Pseudodifferential	Differential
E and M Series, NI PCI-6010, NI 9205, and NI 9206	For devices with eight channels: differential for the first four channels, referenced single-ended for the next four channels. For devices with 16 channels or more: differential for eight channels followed by referenced single-ended for eight channels. For instance, channels 1-7, 16-23, and 32-39 are differential. Channels 8-15, 24-31, and 40-47 are referenced-single ended.	Referenced single-ended
NI PXI-6132, NI PXI-6133, NI PXI-6143	Differential	N/A
NI PCI-6110, NI PCI-6111, NI 6115, NI 6120	Pseudodifferential	Referenced single-ended
NI PXI-42*XX*	Differential	N/A
SCC	Non-referenced single-ended	Referenced single-ended
SCXI	Differential	Differential
NI 9201, NI	Referenced single-ended	N/A

USB-9201, NI 9203, NI 9217, NI 9221, NI USB-9221[1]		
NI 9211, NI USB-9211, NI 9215, NI USB-9215, NI 9219, NI USB-9219, NI 9225, NI 9229, NI USB-9229, NI 9235, NI 9236, NI 9237, NI USB-9237, NI 9239, NI USB-9239[1]	Differential	N/A
NI 9233, NI USB-9233, NI 9234, NI USB-9234	Pseudodifferential	N/A
NI 9263, NI USB-9263, NI 9264, and NI 9265	N/A	Referenced single-ended
[1] All listed devices have a fixed terminal configuration.		

Terminal Configurations (Analog Input Ground Reference Settings) for Isolated Devices

You can use differential, referenced single-ended (RSE), or nonreferenced single-ended (NRSE) terminal configurations (or analog input ground reference settings) for isolated devices.

The following figure shows a differential measurement system. For illustrations of other terminal configurations, refer to your device documentation.



M Series Isolated Device Configured in DIFF Mode

RSE and NRSE measurement systems are the same for isolated devices in that the measurement is made with respect to a floating or isolated ground, AI GND. AI GND is the floating reference for all RSE and NRSE channels. AI GND is isolated from earth ground through an isolation barrier on the device.

It is important to keep within the specifications of your device to avoid hazardous conditions. It is considered improper use of the device to surpass the specifications, and the device is no longer considered to be in safe use. Isolated devices specify a continuous working isolation voltage that specifies the maximum voltage difference allowed between any of the input signals to the chassis/earth ground. For example, a product rated for 60 VDC of continuous working isolation that has a

voltage difference of +51 VDC between AIGND and the chassis/earth ground cannot have a signal greater than +9 VDC when referenced to a AIGND or 60 VDC when reference to chassis/earth ground at its input terminals.

Routing

This section contains information about routing for AO Series, E Series, S Series, and TIO devices.

Routing Considerations for AO Series Devices

The following sections detail special routing considerations for AO Series devices.

Counters

The counters on AO Series devices are very versatile and in many cases can route signals across subsystems. They can also be used to route signals to/from the I/O connector. However, when a counter is used as part of a route, you may not be able to use the counter for other applications while the route remains reserved. Most routes do not require an internal counter terminal, but many advanced routes do. For example, if you want to use the signal present at PFI 4 on Dev1 as the Start Trigger for an acquisition on Dev2, you simply need to specify /Dev1/PFI4 as the source of the trigger. However, to make the route, the signal is internally routed from /Dev1/PFI4 to /Dev1/Ctr0Source to a RTSI bus line or PXI_Trig to Dev2/ai/StartTrigger. These terminals need not be explicitly specified when programming the route. In this case, it is not obvious that a counter terminal is used to make the route. Subsequent attempts at using the counter while it is in use result in a routing reservation error. To see if the route you are making uses counter resources, consult the table displayed under the Device Routes tab in MAX.

Routing Considerations for E Series and S Series Devices

The following sections detail special routing considerations for E Series and S Series devices.

PFI 0

When exporting a signal through task-based routing to most PFI terminals, the route is reserved and committed with the task. When the task goes back to a verified state, software resources for the route are released, but the route remains in place in hardware. It remains in place to prevent glitching on the PFI terminal and to prevent any unexpected effects on external circuitry monitoring the signal. However, PFI 0 is an exception to this rule. Because PFI 0 can accept both analog and digital signals, it tristates when the task is not in the committed or running state. This behavior is intended to prevent accidental connections of an analog signal directly to digital circuitry that could damage the device.

 **Note** PFI 0 accepts only digital signals on the NI 6154.

When in use, the analog trigger circuitry takes over the PFI 0 terminal internal to the device. Because of this, you cannot use PFI 0 to route any digital signals when using the analog trigger, regardless of whether you are triggering off of PFI 0 or an analog input channel. If you try to use PFI 0 for digital signals and the analog trigger at the same time, you receive a routing error.

Counters

The counters on E Series and S Series devices are very versatile and in many cases can route signals across subsystems. Counters also can be used to route signals to and from the I/O connector. However, when a counter is used as part of a route, you may not be able to use the counter for other applications while the route remains reserved. Most routes do not require an internal counter terminal, but many advanced routes do. For example, if you want to use the signal present at PFI 4 on Dev1 as the Start Trigger for an acquisition on Dev2, you simply need to specify /Dev1/PFI4 as the source of the trigger. However, to make the route, the signal is internally routed from /Dev1/PFI4 to /Dev1/Ctr0Source to a RTSI bus line or PXI_Trig to Dev2/ai/StartTrigger. These terminals need not be explicitly specified when programming the route. In this case, it is not obvious that a counter terminal is used to make the route. Subsequent attempts at using the counter while it is in use result in a routing reservation error. To see if the route you are making uses counter resources, consult the table displayed under the Device Routes tab in MAX.

Routing Considerations for TIO Devices

Counters

Though TIO counters can receive signals from any of the PFI lines, NI-DAQmx uses internal resources to connect some PFIs to counter inputs. There are some PFI lines that do not use internal resources that are preferred for use with different counter signals.

- **Ctr*n*Aux**—PFI 37, PFI 33, PFI 29, PFI 25, PFI 21, PFI 17, PFI 13, PFI 9
- **Ctr*n*Gate**—PFI 38, PFI 34, PFI 30, PFI 26, PFI 22, PFI 18, PFI 14, PFI 10
- **Ctr*n*Source**—PFI 39, PFI 35, PFI 31, PFI 27, PFI 23, PFI 19, PFI 15, PFI 11

For output, the same rules apply. Though the counter can output on any PFI line, there are a subset of preferred PFIs that do not use internal resources to make the routes.

- **Ctr*n*InternalOutput**—PFI 36, PFI 32, PFI 28, PFI 24, PFI 20, PFI 16, PFI 12, PFI 8

To see if the route you are making uses internal resources, consult the table displayed under the Device Routes tab in MAX.

Switches

This section contains information specific to switch devices about API support and switching capacity, including switching voltage, switching current, and switching power.

API Support for Switch Modules

Switch modules can support any of four different ways to control their relays. You may use the APIs interchangeably, but NI recommends using a single API for each application.

- **Digital Output**—Create your tasks, either programmatically with the Create Channel Digital Output function/VI or interactively through the DAQ Assistant, using the digital output physical channels. Use the digital versions of the Write function/VI to control the relays. Each digital port consists of 32 digital lines, and each line represents a relay on the switch. For example, if a module contains 64 relays, the first 32 are on port 0, and the rest will be on port 1. Writing a 0 to a digital line opens the relay and writing a 1 closes it.
- **Immediate**—The immediate API, supported by all switches, provides a switch channel-based interaction recommended for nonscanning operations. Functions/VIs such as DAQmx Switch Connect and DAQmx Switch Disconnect are considered part of the immediate API.
- **Relay**—The relay API provides a relay-based interaction. Functions/VIs like DAQmx Switch Open Relays and DAQmx Switch Close Relays are considered part of the relay API.
- **Scanning**—Scanning is a method of connecting channels and is often used when connecting instruments and devices under test (DUTs) in a specific order. In this operation mode, the switch cycles through each entry in a scan list downloaded to the switch. The triggers the switch receives initiate this cycling. Create scanning tasks using DAQmx Switch Create Scan List and control tasks using functions/VIs like DAQmx Start, DAQmx Stop, and so on.

Supported Topologies

Every switch module supports one or more topologies. Changing the topology alters the functionality of the switch and, in many cases, changes the list of supported channel names.

Special Considerations

Some switch modules have specific behaviors that you must consider when developing applications, described in the following table.

Device	Supported APIs	Supported Topologies
PXI-250124-Channel FET Multiplexer/Matrix	Immediate Relay Scanning	2501/1-Wire 48x1 Mux 2501/1-Wire 48x1 Amplified Mux 2501/2-Wire 24x1 Mux 2501/2-Wire 24x1 Amplified Mux 2501/2-Wire Dual 12x1 Mux 2501/2-Wire Quad 6x1 Mux 2501/2-Wire 4x6 Matrix 2501/4-Wire 12x1 Mux
PXI-2503 24-Channel Relay Multiplexer/Matrix	Immediate Relay Scanning	2503/1-Wire 48x1 Mux 2503/2-Wire 24x1 Mux 2503/2-Wire Dual 12x1 Mux 2503/2-Wire Quad 6x1 Mux 2503/2-Wire 4x6 Matrix 2503/4-Wire 12x1 Mux
PXI-2527 32-Channel 300 V Multiplexer	Immediate Relay Scanning	2527/1-Wire 64x1 Mux 2527/1-Wire Dual 32x1 Mux 2527/2-Wire 32x1 Mux 2527/2-Wire Dual 16x1 Mux 2527/4-Wire 16x1 Mux 2527/Independent
PXI-2529	Immediate	2529/2-Wire 8x16

128-Crosspoint Relay Matrix	Relay Scanning	Matrix 2529/2-Wire 4x32 Matrix 2529/2-Wire Dual 4x16 Matrix
PXI-2530 128-Channel Reed Relay Multiplexer/Matrix	Immediate Relay Scanning	2530/1-Wire 128x1 Mux 2530/1-Wire Dual 64x1 Mux 2530/2-Wire 64x1 Mux 2530/4-Wire 32x1 Mux 2530/1-Wire 4x32 Matrix 2530/1-Wire 8x16 Matrix 2530/1-Wire Octal 16x1 Mux 2530/1-Wire Quad 32x1 Mux 2530/2-Wire 4x16 Matrix 2530/2-Wire Dual 32x1 Mux 2530/2-Wire Quad 16x1 Mux 2530/4-Wire Dual 16x1 Mux 2530/Independent
PXI-2532 512-Crosspoint Matrix	Immediate Relay Scanning	2532/1-Wire 16x32 Matrix 2532/1-Wire 4x128 Matrix 2532/1-Wire 8x64 Matrix 2532/1-Wire Dual 16x16 Matrix 2532/1-Wire Dual 4x64 Matrix

		2532/1-Wire Dual 8x32 Matrix 2532/1-Wire Sixteen 2x16 Matrix 2532/2-Wire 16x16 Matrix 2532/2-Wire 4x64 Matrix 2532/2-Wire 8x32 Matrix
PXI-2533 256-Crosspoint SSR Matrix	Immediate Relay Scanning	2533/1-Wire 4x64 Matrix
PXI-2534 256-Crosspoint SSR Matrix	Immediate Relay Scanning	2534/1-Wire 8x32 Matrix
PXI-2535 544-Crosspoint FET Matrix	Immediate Relay Scanning	2535/1-Wire 4x136 Matrix
PXI-2536 544-Crosspoint FET Matrix	Immediate Relay Scanning	2536/1-Wire 8x68 Matrix
PXI-2545 2.7 GHz 4x1 Terminated 50 Ohm Multiplexer	Immediate Relay Scanning	2545/4x1 Terminated Mux
PXI-2546 2.7 GHz Dual 4x1 50 Ohm Multiplexer	Immediate Relay Scanning	2546/Dual 4x1 Mux
PXI-2547 2.7 GHz 8x1 50 Ohm Multiplexer	Immediate Relay Scanning	2547/8x1 Mux
PXI-2548 2.7 GHz 4-SPDT 50 Ohm Relay Module	Digital Output Immediate Relay Scanning	2548/4-SPDT

PXI-2549 2.7 GHz Terminated 2-SPDT 50 Ohm Relay Module	Digital Output Immediate Relay Scanning	2549/Terminated 2-SPDT
PXI-2554 2.5 GHz 4x1 75 Ohm Multiplexer	Immediate Relay Scanning	2554/4x1 Mux
PXI-2555 2.5 GHz 4x1 Terminated 75 Ohm Multiplexer	Immediate Relay Scanning	2555/4x1 Terminated Mux
PXI-2556 2.5 GHz Dual 4x1 75 Ohm Multiplexer	Immediate Relay Scanning	2556/Dual 4x1 Mux
PXI-2557 2.5 GHz 8x1 75 Ohm Multiplexer	Immediate Relay Scanning	2557/8x1 Mux
PXI-2558 2.5 GHz 4-SPDT 75 Ohm Relay Module	Digital Output Immediate Relay Scanning	2558/4-SPDT
PXI-2559 2.5 GHz Terminated 2-SPDT 75 Ohm Relay Module	Digital Output Immediate Relay Scanning	2559/Terminated 2-SPDT
PXI-2564 16-SPST Relay Module	Digital Output Immediate Relay Scanning	2564/8-DPST 2564/16-SPST
PXI-2565 16-SPST Power Relay Module	Digital Output Immediate Relay	2565/16-SPST

	Scanning	
PXI-2566 16-SPDT Relay Module	Digital Output Immediate Relay Scanning	2566/8-DPDT 2566/16-SPDT
PXI-2567 64-Channel Relay Driver Module	Digital Output Immediate Relay Scanning	2567/Independent
PXI-2568 31-Channel SPST Relay Module	Digital Output Immediate Relay Scanning	2568/15-DPST 2568/31-SPST
PXI-2569 100-Channel SPST Relay Module	Digital Output Immediate Relay Scanning	2569/50-DPST 2569/100-SPST
PXI-2570 40-Channel SPDT Relay Module	Digital Output Immediate Relay Scanning	2570/20-DPDT 2570/40-SPDT
PXI-2575 196x1 Relay Multiplexer	Immediate Relay Scanning	2575/1-Wire 196x1 Mux 2575/2-Wire 98x1 Mux 2575/2-Wire 95x1 Mux
PXI-2576 Multi-Bank Multiplexer	Immediate Relay Scanning	2576/2-Wire Octal 8x1 Mux 2576/2-Wire Sixteen 4x1 Mux
PXI-2584 High-Voltage Multiplexer	Immediate Relay	2584/Independent 2584/1-Wire 12x1 Mux

	Scanning	2584/1-Wire Dual 6x1 Mux 2584/2-Wire 6x1 Mux
PXI-2585 10-Channel Multiplexer	Immediate Relay Scanning	2585/1-Wire 10x1 Mux
PXI-2586 10-Channel SPST Relay Module	Digital Output Immediate Relay Scanning	2586/5-DPST 2586/10-SPST
PXI-2590 1.3 GHz 4x1 50 Ohm Multiplexer	Immediate Relay Scanning	2590/4x1 Mux
PXI-2591 4 GHz 4x1 50 Ohm Multiplexer	Immediate Scanning	2591/4x1 Mux
PXI-2593 500 MHz Dual 8x1 50 Ohm Multiplexer	Immediate Relay Scanning	2593/16x1 Mux 2593/Dual 8x1 Mux 2593/8x1 Terminated Mux 2593/Dual 4x1 Terminated Mux 2593/Independent
PXI-2594 1x4 2.5 GHz Multiplexer	Immediate Relay Scanning	2594/4x1 Mux
PXI-2595 1x4 5.5 GHz Multiplexer	Immediate Relay Scanning	2595/4x1 Mux
PXI-2596 Dual 1x6 26.5 GHz Multiplexer	Immediate Relay Scanning	2596/Dual 6x1 Mux
PXI-2597 1x6 26.5 GHz Terminated Multiplexer	Immediate Relay Scanning	2597/6x1 Terminated Mux

PXI-2598 Dual 26.5 GHz Transfer Switch	Immediate Relay Scanning	2598/Dual Transfer
PXI-2599 Dual 26.5 GHz SPDT	Immediate Relay Scanning	2599/2-SPDT
SCXI-1127 32-Channel Relay Multiplexer/Matrix See SCXI-1127 Considerations	Immediate Relay Scanning	1127/1-Wire 64x1 Mux 1127/2-Wire 32x1 Mux 1127/4-Wire 16x1 Mux 1127/2-Wire 4x8 Matrix
SCXI-1128 32-Channel Solid-State Relay (SSR) Multiplexer/Matrix See SCXI-1128 Considerations	Immediate Relay Scanning	1128/1-Wire 64x1 Mux 1128/2-Wire 32x1 Mux 1128/4-Wire 16x1 Mux 1128/2-Wire 4x8 Matrix 1128/Independent
SCXI-1129 256-Crosspoint Relay Matrix	Immediate Relay Scanning	1129/2-Wire 16x16 Matrix 1129/2-Wire 8x32 Matrix 1129/2-Wire 4x64 Matrix 1129/2-Wire Dual 8x16 Matrix 1129/2-Wire Dual 4x32 Matrix 1129/2-Wire Quad 4x16 Matrix
SCXI-1130 256-Channel Reed Relay Multiplexer/Matrix	Immediate Relay Scanning	1130/1-Wire 256x1 Mux 1130/1-Wire Dual 128x1 Mux 1130/2-Wire 128x1 Mux 1130/4-Wire 64x1 Mux 1130/1-Wire 4x64 Matrix 1130/1-Wire 8x32

		Matrix 1130/1-Wire Octal 32x1 Mux 1130/1-Wire Quad 64x1 Mux 1130/1-Wire Sixteen 16x1 Mux 1130/2-Wire 4x32 Matrix 1130/2-Wire Octal 16x1 Mux 1130/2-Wire Quad 32x1 Mux 1130/4-Wire Quad 16x1 Mux 1130/Independent
SCXI-1160 16-SPDT General-Purpose Relay Module	Digital Output Immediate Relay	1160/16-SPDT
SCXI-1161 8-SPDT Power Relay Module	Digital Output Immediate Relay	1161/8-SPDT
SCXI-1163R 32-Channel SSR	Digital Output Immediate Relay	1163R/Octal 4x1 Mux
SCXI-1166 32-SPDT Relay Module	Digital Output Immediate Relay Scanning	1166/16-DPDT 1166/32-SPDT
SCXI-1167 64-Channel Relay Driver Module	Digital Output Immediate Relay	1167/Independent

	Scanning	
SCXI-1169 100-Channel SPST Relay Module	Digital Output Immediate Relay Scanning	1169/50-DPST 1169/100-SPST
SCXI-1175 196x1 Relay Multiplexer	Immediate Relay Scanning	1175/1-Wire 196x1 Mux 1175/2-Wire 98x1 Mux 1175/2-Wire 95x1 Mux
SCXI-1190 1.3 GHz Quad 4x1 50 Ohm Multiplexer	Immediate	1190/Quad 4x1 Mux
SCXI-1191 4 GHz Quad 4x1 50 Ohm Multiplexer	Immediate	1191/Quad 4x1 Mux
SCXI-1192 18 GHz 8-SPDT 50 Ohm Relay Module	Digital Output Immediate Relay	1192/8-SPDT
SCXI-1193 500 MHz Quad 8x1 50 Ohm Multiplexer	Immediate Relay Scanning	1193/32x1 Mux 1193/Dual 16x1 Mux 1193/Quad 8x1 Mux 1193/16x1 Terminated Mux 1193/Dual 8x1 Terminated Mux 1193/Quad 4x1 Terminated Mux 1193/Independent
SCXI-1194 Quad 1x4 2.5 GHz Multiplexer	Immediate Relay Scanning	1194/Quad 4x1 Mux
SCXI-1195 Quad 1x4 5.5 GHz Multiplexer	Immediate Relay Scanning	1195/Quad 4x1 Mux

SCXI-1127 Considerations

To route signals to the analog bus backplane, you must enable the switch device property **Auto Connect Analog Bus**. As a result, if you connect a channel (ch1) to the common channel (com0), the signal is automatically routed from com0 to the analog bus (ab0).

The SCXI-1127 supports only continuous scanning.

If you have used immediate or relay operations to change relay states before starting a scan, all of those relays are opened when the scan starts. After the scan completes, the relays are returned to their previous state prior to the scan.

Early revisions of this hardware reserve the SCXI_TRIG1 line. If you place an older revision of this hardware (earlier than revision E) into an SCXI chassis that also contains an SCXI analog input module that performs track and hold (such as the SCXI-1140 or SCXI-1520), you may get reservation errors when trying to use the SCXI analog input module.

SCXI-1128 Considerations

To route signals to the analog bus backplane, you must enable the switch device attribute/property **Auto Connect Analog Bus**. As a result, if you connect a channel (ch1) to the common channel (com0), the signal is automatically routed from com0 to the analog bus (ab0).

The SCXI-1128 supports only continuous scanning.

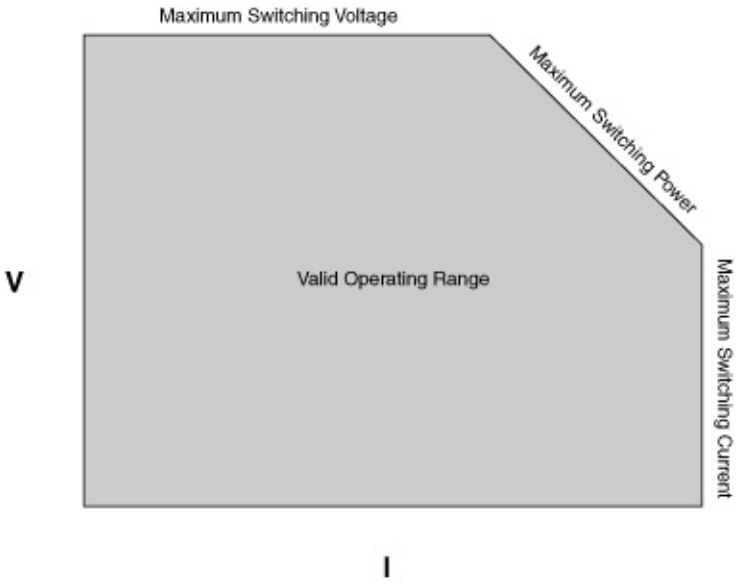
If you have used immediate or relay operations to change relay states before starting a scan, all of those relays are opened when the scan starts. After the scan completes, the relays are returned to their previous state prior to the scan.

Switching Capacity

Signal levels through a switch must account for the following specifications:

- Switching voltage
- Switching current
- Switching power

The following figure shows the valid operating range defined by these limits.



Switching Current

Switching current is the maximum rated current that can flow through the switch as it makes or breaks a contact. Switching active currents results in arcing that can damage the contacts of electromechanical relays. A minimum current specification indicates the smallest current that can reliably flow through the switch.

Switching Power

Switching power is the limit on the combined open-contact voltage and closed-contact current of a signal in the switch.

Switching Power = Switching Voltage * Switching Current

Switching high-power signals causes high-energy arcing at the electromechanical contacts during actuation, reducing the useful life of the switch.

Switching Voltage

Switching voltage refers to the maximum signal voltage that the switch module can safely maintain. Switching voltage is defined from channel-to-ground and from channel-to-channel. Channel-to-ground is the voltage potential between the signal line and the grounded chassis. Channel-to-channel is the voltage potential between any pair of signal lines within the module. This voltage includes voltages across open relay contacts, as well as voltages between adjacent connection terminals.

Note CE marking for measurement and control devices requires compliance to the IEC 61010-1 standard. Switch modules intended for high-voltage signals (> 60 VDC / 30 V_{rms}) are rated for Measurement Categories as defined in this standard. Measurement Categories describe the acceptable transient overvoltages and fault protection necessary for safe operation. Refer to the *NI Switches Getting Started Guide* for more information on Measurement Categories.

Synchronization

This section contains information on synchronizing multiple E Series devices as well as multiple DSA devices.

E Series

DSA

M Series USB

Synchronizing DSA Devices

Note If you want to synchronize analog input on two or more DSA devices at the same sampling rate, you can use channels from those devices within the same task.

You can synchronize the analog input and output operations on two or more DSA devices to extend the channel count of DSA measurements. Two synchronization methods are available, depending on the types of devices you want to synchronize and your overall system configuration:

- **Sample Clock Timebase Synchronization**—Using this method, you can synchronize any combination of NI 446*X* and NI 447*X* devices. The master device exports its sample clock timebase to the slave devices and must be in slot 2 for PXI devices. On PXI devices, you can synchronize up to 14 devices per chassis.
- **Reference Clock Synchronization**—This method can be used to synchronize NI PXI-446*X* devices only. Both master and slave devices lock the sample clock timebase to the shared 10 MHz reference clock on the PXI chassis backplane. Master and slave devices can be in any slot, and devices in all slots in the PXI chassis may be synchronized.

For more information on synchronization for DSA devices, refer to the following sections:

- Signals for Synchronizing DSA Devices—This section introduces the signals required to synchronize DSA devices and briefly discusses the programming steps needed to share them.
- Homogeneous DSA Device Synchronization—This section contains information about synchronizing the same types of DSA devices at the same sampling rate.
- Heterogeneous DSA Device Synchronization—This section contains information on multirate applications and synchronizing different types of DSA devices, such as an NI 446*X* with an NI 447*X*.
- Multi-Chassis DSA Synchronization—This section covers configuring a synchronized DSA application using more than one PXI chassis.
- DSA Synchronization Examples—This section contains

descriptions of example programs that illustrate synchronization.

DSA Synchronization Examples

The example program Multi-Device Sync AI-Shared Timebase & Trig-DSA VI provides a simple example for synchronizing two DSA devices for analog input. The Multi-Device Sync-AI and AO-Shared Timebase & Trig-DSA VI illustrates the steps to synchronize two DSA devices for both analog input and output operations.

Heterogeneous DSA Device Synchronization

A synchronized DSA system can consist of NI 446*X* devices, NI 447*X* devices, or a combination thereof. Heterogeneous synchronization refers to a synchronized system with disparate devices or disparate device settings. If any of the following conditions are true, the synchronization is considered to be heterogeneous:

- Synchronizing NI 446*X* devices with NI 447*X* devices.
- Synchronizing only NI 446*X* devices at different sampling rates or only NI 447*X* devices at different sampling rates.

Synchronizing NI 446X and NI 447X Devices

The following constraints apply when synchronizing NI 446*X* and NI 447*X* devices:

- For PXI devices, the master device must be an NI 446*X*, and that device must be in slot 2.
- Only the sample clock timebase synchronization method can be used.
- When using different sampling rates on the devices, the sampling rates on all devices must be greater than 25.6 kS/s.
- When using different sampling rates on the devices, the ratio between the sampling rates must be a power of two.
- You must disable enhanced alias rejection on all devices.
- You must account for filter delay differences between the devices.

When programming, the implementation of synchronizing NI 446*X* devices and NI 447*X* devices is the same as the homogeneous sample clock timebase synchronization case. Refer to Signals for Synchronizing DSA Devices.

Synchronizing NI 446*X* Devices or NI 447*X* Devices at Different Sampling Rates

The following constraints apply when synchronizing only NI 446*X* devices at different sampling rates or only NI 447*X* devices at different sampling rates and using sample clock timebase synchronization:

- The ratio between sampling rates on all devices must be a power of two.
- NI 447*X* devices must use a sampling rate greater than 25.6 kS/s.

There are no constraints for multirate applications when using reference clock synchronization on supported devices.

The following are programming caveats to use when synchronizing only NI 446*X* devices at different sampling rates or only NI 447*X* devices at different sampling rates:

- When using sample clock timebase synchronization, account for differences in expected Sample Clock Timebase frequencies. First, read the **SampClk.Timebase.Rate** attribute/property from the master device. Then write this value to the **SampClk.Timebase.Rate** attribute/property on each slave.
- Regardless of synchronization method, ensure the master device allows the Sync Pulse sufficient time to reset all ADCs and DACs in the system. To guarantee adequate sync time, read the **SyncPulse.SyncTime** attribute/property from all slave devices. Then write the maximum of these values to the **SyncPulse.MinDelayToStart** attribute/property on the master before committing the master task.
- Regardless of synchronization method, you must account for filter delay between the devices.

Homogeneous DSA Device Synchronization

Homogeneous DSA synchronization is defined as synchronizing DSA devices in a system that meets both of the following criteria:

- The set of synchronized devices consists of either all 447*X* devices or all 446*X* devices, but not a combination of 447*X* and 446*X* devices.
- All synchronized devices run at the same sample rate.
- If those conditions are not met, the devices are in a heterogeneous synchronization system.

Multiple signals must be shared in a synchronized DSA system.

Multi-Chassis DSA Synchronization

You can synchronize up to 14 PXI devices in a single PXI chassis using sample clock timebase synchronization. You can synchronize up to 17 PXI devices in a single PXI chassis using reference clock synchronization. You can also synchronize DSA operations among multiple PXI chassis. Multi-chassis DSA synchronization requires that each PXI chassis include an NI PXI-665*X*. The NI PXI-665*X* devices must be programmed to share the Sync Pulse, the Start Trigger, and either the Sample Clock Timebase or Reference Clock between chassis, depending on the desired synchronization method. The software for the NI PXI-665*X* includes several example programs illustrating multi-chassis DSA synchronization.

Signals for Synchronizing DSA Devices

For a sample clock timebase synchronized DSA system, the Sample Clock Timebase, Sync Pulse, and Start Trigger signals must be shared between the master and slave devices. The master device is the source of all three signals.

Note If you are developing a synchronized DSA application using PXI devices, and if you are using sample clock timebase synchronization, the master DSA device must reside in slot 2 of the PXI chassis.

For a reference clock synchronized DSA system, the Sync Pulse and Start Trigger signals must be shared between the master and the slave devices, and all devices must specify PXI_Clk10 as the reference clock source. The master device is the source of the Sync Pulse and Start Trigger.

Sample Clock Timebase

In a PXI system, the master device exports the Sample Clock Timebase to one or more PXI_Star lines. For PCI DSA devices, the Sample Clock Timebase can only be exported or imported on RTSI 8. Other devices, such as E Series devices, cannot access this signal. The frequency of the Sample Clock Timebase depends on the desired sampling rate and on the DSA device, but in every case, this signal is many times faster than the desired sampling rate. The slave DSA devices individually divide the Sample Clock Timebase signal internally to produce their sample clocks. You can access the Sample Clock Timebase signal with the Sample Clock **Source** attribute/property.

Sync Pulse

The Sync Pulse simultaneously resets the internal clock dividers and converters on each DSA device in the system. This eliminates any phase difference on the Sample Clock Timebase dividers on each device to guarantee tight phase matching across input and output channels in the system. In NI PXI 447*X* devices, this signal must be routed along PXI_Trig5. In NI PCI 447*X* devices, the Sync Pulse must be routed along RTSI 9. In NI 446*X* devices, you can use any RTSI or PXI_Trig line from RTSI0::6 or PXI_Trig0::6. You can program the Sync Pulse routing with the **SyncPulse.Src** attribute/property.

The Sync Pulse is not sent until you commit the master task. Starting a task also commits it. The slave task must be committed before the master task. If doing analog output, the Write function/VI commits the task. You must call this function/VI on the slave task before the master. The converter reset operation that follows the Sync Pulse requires some time, from several milliseconds to several seconds, depending on the sampling rate and specific DSA devices in the system. This reset time must elapse before the acquisition begins. The reset delay is not present in single-device DSA systems. In general, the delay is noticeable only at sampling rates below about 10 kS/s.

Note If you set the Sync Pulse source on a task to its own Sync Pulse signal, that task will be configured as a slave task. You must not program the **SyncPulse.Src** attribute/property unless you want the task to be programmed as a slave task.

Start Trigger

You should program each slave device for digital triggering using the appropriate RTSI or PXI_Trig line as the trigger source. The master device can export this signal on RTSI/PXI_Trig0::4 (NI 447*X* devices) or RTSI/PXI_Trig0::6 (NI 446*X* devices).

Reference Clock

When using reference clock synchronization, the sample clock timebase is not shared between master and slave tasks. Instead, all devices lock their onboard sample clock timebase to a shared 10 MHz signal on the PXI chassis backplane. You can program the reference clock source with the **RefClk.Src** attribute/property. The syntax for this is PXI Clk_10.

Synchronizing E Series Devices

This section covers the signals commonly used for synchronizing E Series devices.

Analog Input Tasks

If you are synchronizing analog input tasks, you can do so by sharing the following signals:

- **Master Timebase and a Start Trigger**—All synchronized devices are programmed to use the same signal (usually the 20MHzTimebase from one of the devices) as their Master Timebase. More generally, one device can be queried for its Master Timebase source and that terminal can be set as the source of the Master Timebase for the other synchronized devices. All synchronized devices are programmed to use the same ai/StartTrigger terminal as the source of their Start Trigger. You can always share a Start Trigger even if you have not explicitly configured one. There are two advantages of using this method. The devices need not sample at the same rate, nor acquire the same amount of data. This method also works for synchronizing analog output signals and some counter applications. The disadvantages of using this method are two signals need to be routed using two RTSI or PXI trigger lines. An attribute/property must be set to designate the 20MhzTimebase of another device as the Master Timebase of a synchronized device.
- **AI Sample Clocks**—All synchronized devices are programmed to use the ai/SampleClock terminal from one of the devices as their Sample Clock. The advantage of using this method is only a single signal is routed using but one RTSI line or PXI Trigger line. The disadvantages are that all devices must sample at the same rate and can acquire no more data than the device that is sourcing the AI Sample Clock.

Analog Output Tasks

If you are synchronizing analog output tasks, you can do so by sharing the following signals:

- **Master Timebase and a Start Trigger**—All synchronized devices are programmed to use the same signal (usually the 20MHzTimebase from one of the devices) as their Master Timebase. More generally, one device can be queried for its Master Timebase source and that terminal can be set as the source of the Master Timebase for the other synchronized devices. All synchronized devices are programmed to use the same ao/StartTrigger terminal as the source of their Start Trigger. You can always share a Start Trigger even if you have not explicitly configured one. The advantages of using this method are that the devices need not generate samples at the same rate, nor generate the same amount of data. This method also works for synchronizing analog input signals and some counter applications. The disadvantage of using this method is two signals need to be routed using two RTSI or PXI Trigger lines. An attribute/property must be set to designate the 20MhzTimebase of another device as the Master Timebase of a synchronized device.
- **AO Sample Clocks**—All synchronized devices are programmed to use the ao/SampleClock terminal from one of the devices as their Sample Clock. The advantages of using this method are only a single signal is routed using one RTSI line or PXI Trigger line. The disadvantages are that all devices must generate samples at the same rate and can generate no more data than the device that is sourcing the AO Sample Clock.

Counter Tasks

You cannot synchronize counter input applications performing period, frequency, pulse width or semi-period measurements in the same sense as analog input or output applications. These types of counter input applications cannot be programmed to make their measurements at the same time because the signals being measured themselves determine when the measurements are made, and there is no reason to set up multiple devices to measure the same signal. You also cannot use Start Triggers for counter input applications.

You can, however, ensure that all counters are using the same timebase for their input measurements by sharing the CI Counter Timebase signal. Program all devices to use the same signal (usually the 20MHzTimebase from one of the devices) as their CI Counter Timebase. More generally, one device can be queried for its CI Counter Timebase source and that terminal can be set as the source of the CI Counter Timebase for the other devices.

If you are synchronizing buffered counter input applications performing edge counting, you can do so by sharing the Sample Clock. The Sample Clock must be externally supplied to one of your devices. The other synchronized devices are programmed to use this device's Ctr*n*Gate signal as their Sample Clock, where *n* is the number of the counter.

If you are synchronizing pulse generation counter output applications, you can do so by sharing the CO Counter Timebase and Start Trigger signal. Program all devices to use the same signal (usually the 20MHzTimebase from one of the devices) as their CO Counter Timebase. More generally, one device can be queried for its CO Counter Timebase source and that terminal can be set as the source of the CO Counter Timebase for the other devices. Program all devices to use the same signal as their Digital Edge Start Trigger. This is typically the Ctr*n*Gate signal from one of the devices, where *n* is the number of the counter.

Other E Series Synchronization Notes

PXI E Series devices use PXI_Trig 6 to receive PXI Star trigger signals. Thus, these PXI E Series devices cannot use PXI_Trig 6 to communicate with one another.

PCI E Series devices import 20 MHz Timebase clock only from RTSI 7. PXI E Series devices import 20 MHz Timebase clock only from PXI_Trig 7.

PXI_CLK 10 with the NI PXI-6608

When NI-DAQmx detects that an NI PXI-6608 is installed in slot 2, NI-DAQmx automatically overrides PXI_CLK10 with a more stable oven-controlled oscillator (OCXO) when NI-DAQmx loads. If you switch between Traditional NI-DAQ (Legacy) and NI-DAQmx, both drivers continue to override the PXI_CLK10 with the OCXO.

Refer to the NI PXI-6608 documentation for more information about the stability of the OCXO.

Note For the automatic override to occur in NI-DAQmx, you must put the NI PXI-6608 in slot 2. For more information about configuring your PXI chassis, refer to the *Measurement & Automation Explorer Help for PXI*.

Synchronization with M Series USB Devices

M Series USB devices do not support reference clock synchronization.

Timing

This section contains information about timing for AO Series, C Series, DSA, E Series, M Series USB, and S Series devices.

Timing Considerations for AO Series Devices

When using an external ao/SampleClock for finite generations, you need to provide one more sample clock pulse than the number of samples in the generation. The Wait Until Done function/VI uses the extra sample clock to indicate the task is complete. For example, if you want to generate 1000 samples using an external sample clock, the first 1000 samples clocks you provide generates all of the samples, but you need to provide 1001 sample clock pulses for the Wait Until Done function/VI to indicate the task is done. If you are trying to synchronize an analog output generation with another acquisition or generation by sharing a common clock, use the ao/SampleClock as the master clock, or key off of the generation or acquisition providing the master clock to determine when the generation is complete.

Static AO devices, such as the NI 6703 and NI 6704, do not have hardware timing and have multiplexed output. Refer to your device documentation for specifics concerning your device.

Timing Considerations for C Series Devices

Note C Series devices do not support hardware-timed single-point sample mode or Wait for Next Sample Clock.

Analog Input

You can use multiple analog input devices of different types in the same task, and NI-DAQmx automatically synchronizes them as long as they are in the same chassis. NI-DAQmx supports only one analog input task at a time per cDAQ chassis.

AI Convert Clock Considerations

The NI 9201, NI 9203, NI 9205, NI 9206, NI 9211, NI 9217, and NI 9221 use multiplexed sampling controlled by a per-slot AI Convert Clock. If you have multiple devices in one task, their AI Convert Clocks run in parallel, which may cause channels on multiple devices to be sampled at the same time. You can set the AI Convert Rate and the Delay From Sample Clock differently on each device. When setting **AI Convert** timing attributes/properties, you must use the **ActiveDevs** attribute/property to specify the device to which you are referring. External clocking of the AI Convert Clock is not supported.

The default AI Convert Clock rate for the NI 9201, NI 9203, NI 9205, NI 9206, NI 9211, NI 9217, and NI 9221 uses 10 µs of additional settling time between channels, compared to the fastest AI Convert Clock rate for the device. When the Sample Clock rate is too high to allow for 10 µs of additional settling time, the default AI Convert Clock rate uses as much settling time as is allowed by the Sample Clock rate. If there are multiple NI 9201, NI 9203, NI 9205, NI 9206, NI 9211, NI 9217, and NI 9221 devices in the same task, the same amount of additional settling time is used for all devices in the task, even if their maximum AI Convert Clock rates differ.

Sampling Rate Considerations

With the NI 9211, NI 9217, and NI 9219, if the sampling rate of a hardware-timed acquisition exceeds the maximum sampling rate of the module, the most recently acquired sample may be read multiple times and no warning or error is generated. Exceeding the maximum sampling rate of other devices in the same task generates warnings or errors. The first sample of a hardware-timed acquisition with the NI 9211, NI 9217, and NI 9219 is sampled when the task is committed. Software-timed acquisitions with the NI 9211, NI 9217, and NI 9219 always wait for a new sample to be acquired.

The maximum sampling rate of the NI 9215 depends on which channel(s)

you are acquiring from. For instance, a task acquiring from any combination of ai0, ai1, and ai2 can sample at faster rates than a task that includes ai3. The maximum sampling rates are attainable only when sampling from ai0. If you have multiple NI 9215 devices in the same task, they sample in parallel. For instance, multiple NI 9215 devices acquiring from ai0 may be able to achieve a faster sampling rate than a single NI 9215 acquiring from ai3.

The NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239 have both a maximum and a minimum sampling rate. Refer to the specifications for your device to determine the sampling rate range.

When the NI 9234 is in a task with a C Series device that has a different sample clock timebase, NI-DAQmx always chooses the sample clock timebase with the highest frequency. To override this selection, you can set the sample clock timebase in the **Sample Clock Timebase Source** attribute/property.

Hardware and On-Demand Timing for the NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239

The NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239 do not support the on-demand timing type. All NI 9225, NI 9229, NI 9233, NI 9234, NI 9237, and NI 9239 acquisitions and generations require hardware timing from a steady clock. You cannot set the **Sample Clock Source** attribute/property to an external source when an NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, or NI 9239 is in the task. With the NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239, external clocking from arbitrary external signal sources such as encoders and tachometers is not supported.

Analog Output

Only one hardware-timed analog output task per chassis at a given time is supported, but the number of concurrent software-timed analog output tasks is limited only by the available channels. A single C Series analog output device cannot be used for hardware-timed and software-timed tasks at the same time.

Digital Input/Output

Only one hardware-timed digital input task and one hardware-timed digital output task per chassis at a given time is supported, but the number of concurrent software-timed digital I/O tasks is limited only by the available lines.

Hardware-timed digital input/output is only supported on correlated digital I/O modules. Refer to Digital I/O Considerations for C Series for more information.

Configurable ADC Timing

On the NI 9217, NI 9219, and NI USB-9219, you can configure high speed, high resolution, or low noise measurements using the **ADC Timing Mode** attribute/property. For the NI 9217 and NI 9219 in CompactDAQ, this attribute/property is set to High Resolution by default. To increase the conversion rate, set this attribute/property to High Speed. For the NI USB-9219, this attribute/property is set to High Resolution by default in on-demand mode, and the default value in hardware-timed mode is automatically determined based on **Sample Clock Rate**. To increase power line noise rejection on the NI 9219, set this attribute/property to Best 60 Hz Rejection or Best 50 Hz Rejection.

On the NI 9217, the **ADC Timing Mode** attribute/property affects both the maximum and default values for **AI Convert Rate** attribute/property in the DAQmx Timing property node. For instance, if the ADC timing mode corresponds to a conversion time of 200 ms, the maximum convert rate is 5 Hz.

Note To set the set the convert clock rate for each device, you need to specify the active devices in the task and specify the **AI Convert Rate** attribute/property.

The ADC timing mode must be the same for all channels on the module, but may differ on multiple modules in the same task. On the NI 9219 and NI USB-9219, using a thermocouple or CJC channel increases the conversion time by 10 ms. Refer to your device documentation for the specific conversion rates.

Timing Considerations for DSA Devices

Supported Sampling Rates

Unlike some other DAQmx devices, DSA devices have both a maximum and a minimum sampling rate. Refer to the specifications for your device to determine the sampling rate range.

Other DSA Timing Considerations

DSA devices do not support the on-demand timing type. All DSA acquisitions and generations require hardware timing from a steady clock.

DSA devices do not support external clocking from arbitrary external signal sources such as encoders and tachometers. The PFI lines on DSA devices cannot accept external clocks. You can program a DSA device to use an external clock only when it is a slave in multi-device synchronized system. Refer to DSA Synchronization for more details.

Timing Considerations for E Series Devices

The following is a list of special timing considerations you should be aware of when using E Series devices:

- **ai/ConvertClock**—When using the ai/ConvertClock as the source of a route, one extra convert pulse is generated than you might expect. For example, if you perform a finite acquisition of 100 samples with four channels, you see 401 convert pulses instead of 400. This extra convert pulse is necessary to set up the configuration memory in hardware and occurs as the task transitions to the committed state.
- **ao/SampleClock**—When using an external ao/SampleClock for finite generations, you need to provide one extra sample clock pulse than the number of samples in the generation for the Wait Until Done function/VI to indicate the task is complete. For example, if you want to generate 1000 samples using an external sample clock, you need to provide 1001 sample clock pulses or the Wait Until Done function/VI never indicates the task is done. All of the samples are generated, but the analog output timing engine needs one additional clock pulse to indicate the generation is complete. If you are trying to synchronize an analog output generation with another acquisition or generation by sharing a common clock, use the ao/SampleClock as the master clock or key off of the generation or acquisition providing the master clock to determine when the generation is complete.

Timing Considerations with M Series USB Devices

M Series USB devices do not support hardware-timed single-point sample mode or Wait for Next Sample Clock.

Timing Considerations for S Series

Analog Input with Pipelined ADCs

Note Not all S Series devices have pipelined ADCs. Refer to the specifications for your device to determine if your device contains pipelined ADCs.

Many S Series devices have pipelined ADCs with an intrinsic pipeline depth. This pipelining allows the device to sample at higher rates, but it also has other consequences on the timing requirements for the device. S Series devices, except for the NI 6143, do not support AI hardware-timed single-point sample mode. Since the data needs to travel through the pipeline before it can be read, the data being read is always pipeline-depth points old. For instance, if the pipeline depth for a device is three, the first sample is acquired on clock tick 1, but it is not available for reading until clock tick 4. Following this logic, you must supply pipeline-depth extra clock pulses for a finite acquisition to flush the pipeline. Continuing with the previous example, if the pipeline depth is three and you want to acquire 1000 samples, you need to generate 1003 sample clock pulses. If you are using the onboard sample clock, NI-DAQmx automatically generates the appropriate number of sample clock pulses. However, when using an external sample clock or when synchronizing devices, you need to ensure you supply the appropriate number of sample clock pulses.

There is also a finite amount of time a sample can be held in the pipeline before it starts to degrade and lose measurement accuracy. This time limit imposes a minimum sampling rate that must be met to achieve the measurement accuracy specified for the device. Although you can sample slower than this minimum recommend sampling rate, the accuracy specifications for the device are not guaranteed. Refer to the specifications for your device to determine the recommended minimum sampling rate.

This degradation of samples in the pipeline also affects on-demand single-point acquisitions and acquisitions that use a Pause Trigger. For on-demand single-point acquisitions, NI-DAQmx generates multiple sample clocks at the maximum sample rate of the device for each sample that is read. For S Series devices with a pipelined ADC, the number of sample clocks generated is equal to the pipeline depth plus one. For S

Series devices that do not have a pipelined ADC, two sample clock pulses are generated for each point. This means that if you export the sample clock while doing an on-demand single-point acquisition, you get more sample clock pulses than data points. NI-DAQmx then throws away all points except the data point that corresponds to the first sample clock pulse. This ensures the data returned is always valid data. For acquisitions that use a Pause Trigger, the trigger could invalidate the samples in the pipeline if the trigger is asserted longer than the pipeline depth divided by the minimum sampling rate. For instance, if the device has a pipeline depth of three and a minimum sampling rate of 1000 samples per second, data should not sit in the pipeline for more than 3 ms. This gives up to a maximum of 3 ms for the Pause Trigger to remain asserted and three sample clocks to be detected before the data in the pipeline deteriorates past specifications. In the case of a Pause Trigger, NI-DAQmx does not detect or throw out any invalid samples. You must detect this situation and deal with any invalid samples as appropriate.

Analog Output

When using an external ao/SampleClock for finite generations, you need to provide one more sample clock pulse than the number of samples in the generation for the Wait Until Done function/VI to indicate the task is complete. For example, if you want to generate 1000 samples using an external sample clock, you need to provide 1001 sample clock pulses, or the Wait Until Done function/VI never indicates the task is done. All of the samples are generated, but the analog output timing engine needs one additional clock pulse to indicate the generation is complete. If you are trying to synchronize an analog output generation with another acquisition or generation by sharing a common clock, use the ao/SampleClock as the master clock, or key off of the generation or acquisition providing the master clock to determine when the generation is complete.

Note For S Series devices that use the STC II timing chip, such as the NI 6154, you do not need to apply an extra sample clock pulse to complete the task.

Sample Rate Considerations

E Series devices have a scanning architecture, which can impact the maximum sample rate in multichannel applications. To find the effective maximum sample rate for these devices, divide the device's sample rate by the number of channels in the task. For instance, if you create a task with 16 channels, and your device has a maximum sample rate of 1 MS/s, the maximum effective sample rate for that task is 62,500 kS/s (1 MS/16 = 62,500 kS).

DSA

This section contains information specific to DSA devices.

Alias Rejection (DSA and C Series)

DSA devices and the NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239 employ a class of ADCs and DACs known as delta-sigma converters. Delta-sigma ADCs include built-in digital filters to provide alias protection from out-of-band signal components. The digital filters always impart a delay of several samples between the time when a given analog voltage level becomes present at the ADC input and when the converter returns the corresponding digitized value. The length of this delay is always deterministic for a particular device running at a given sampling rate.

Likewise, interpolators and delta-sigma DACs provide digital filtering on analog output signals to eliminate out-of-band imaging and quantization noise. As with analog input, the digital output filtering results in a deterministic delay through the DAC.

You can safely ignore the effects of the digital filter delay for most input-only or output-only applications. The filter delay can become significant for applications requiring input and output synchronization such as stimulus-response testing and tight loop control. If your application employs external digital triggering, the acquisition returns data that occurred in time before the trigger event. The number of samples preceding the trigger matches the ADC filter delay. Refer to your device documentation for more details on the ADC and DAC digital filter delays.

Alias Rejection at Low Sample Rates

At very low sample rates, the anti-aliasing filters for AI channels on DSA devices as well as the NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239 may not completely reject all out-of-band signals. This is primarily due to the internal digital filter of the delta-sigma ADC, which cannot suppress signals with frequencies near the multiples of the oversample rate (sampling rate multiplied by oversample factor). These devices also employ fixed cutoff analog lowpass anti-aliasing filters, but at low sample rates, some multiples of the oversample rate can fall below the cutoff frequency of the analog anti-aliasing filter.

For example, for a device using ADCs with an oversample factor of 128X and sampling at a rate of 1 kS/s, the oversample rate is 128 kHz. Some multiples of that oversample rate fall below the cutoff of the analog anti-aliasing filter. If the signal to be digitized contains energy near these frequencies, aliasing can result.

One way to prevent aliasing is to raise the sample rate so that the first 128X multiple of the sample rate falls above the cutoff of the analog anti-aliasing filter. For example, a sample rate of 25.6 kS/s is not subject to aliasing because the first 128X multiple (3.2 MHz) is well above the cutoff frequency of the analog anti-aliasing filter. Some DSA devices support enhanced alias rejection, which automatically handles alias rejection at low sample rates. Refer to the device documentation for the specifics of your device.

Enhanced Alias Rejection

To avoid aliasing at low sample rates, some DSA devices support enhanced alias rejection. With enhanced alias rejection enabled, the device clocks the ADCs at a multiple of the user-specified sample rate. This results in an improvement in low-frequency alias rejection. With enhanced alias rejection enabled, you do not need to scale the desired sample rate, and you do not need to programmatically decimate data the device returns.

Enhanced alias rejection is controlled with the **AI.EnhancedAliasRejectionEnable** attribute/property. Enhanced alias rejection is enabled by default on NI 446*X* devices and disabled by default on NI 447*X* and NI 449*X* devices.

Note The original versions of NI 447*X* devices do not support Enhanced Alias Rejection. Refer to *National Instruments Dynamic Signal Acquisition Help* for more information.

Synchronization Issues

When synchronizing multiple devices, set the **AI.EnhancedAliasRejectionEnable** attribute/property to the same value on all devices. If any of the synchronized devices do not support enhanced alias rejection, set **AI.EnhancedAliasRejectionEnable** to FALSE on all devices. When synchronizing devices from different categories, NI 446*X* with NI 447*X* for example, set **AI.EnhancedAliasRejectionEnable** to FALSE on all devices.

Channel Order

On DSA devices, you must list channels in a task in ascending order. For example, if your task includes ai0 and ai1, you must arrange the channel list such that ai0 precedes ai1.

This constraint applies to virtual channels as well as physical channels. For example, if you include a virtual channel for ai0 named vibration and a virtual channel for ai1 named proxProbe, proxProbe must precede vibration in the channel list.

Gain for DSA Devices

On DSA devices, each gain setting corresponds to a particular range centered on 0 V. The gain settings are specified in decibels (dB), where the 0 dB reference is the default range of ±10 V.

For analog input operations, a negative gain value implies attenuation of the signal before the ADC, increasing the range beyond ±10 V. Thus, an input gain setting of –10 dB corresponds to an input range of ±31.6 V. On analog output, a negative gain value implies attenuation following the DAC. This decreases the output range. For instance, an output gain setting of –20 dB corresponds to an output range of ±1 V.

NI-DAQmx has three separate attribute/property sets you can use to control the hardware gain setting. Each has a different priority. If you write values to two or more of these attributes/properties that correspond to different hardware gain settings, the one with the highest priority will determine the hardware behavior.

- **Gain Attributes/Properties**—The gain attributes/properties **AI.Gain** and **AO.Gain** set the amount of gain to apply to the signal. These properties are set in decibels referenced to 10 V. These properties have the highest priority in NI-DAQmx.
- **Range Attributes/Properties**—The range attributes/properties **AI.Rng.High**, **AI.Rng.Low**, **AO.DAC.Rng.High**, and **AO.DAC.Rng.Low** define the maximum and minimum voltages you can acquire or generate. The range attributes/properties have a lower priority than the gain attributes/properties, but a higher priority than the maximum and minimum attributes/properties.
- **Maximum and Minimum Attributes/Properties**—The maximum and minimum attributes/properties **AI.Max**, **AI.Min**, **AO.Max**, and **AO.Min** specify values in engineering units that define the range. These attributes/properties have the lowest priority in NI-DAQmx. They are also the most commonly used since you can set them immediately when an NI-DAQmx task is created.

Hardware Data Compression

If the raw data compression type is set to lossless packing and all channels in a task support hardware compression, hardware data compression is enabled by default. However, if any channels do not support hardware data compression, software data compression is selected by default.

Integrated Electronic Piezoelectric Excitation (IEPE)

If you attach an IEPE accelerometer or microphone to an AI channel that requires excitation from a DSA device or an NI 9234 , you must enable the IEPE excitation circuitry for that channel to generate the required current. IEPE signal conditioning can be independently configured on a per-channel basis.

To enable the IEPE current source on your DSA device or your NI 9234, use the Channel attribute/property **AI.Excit.Val** to specify a current in amperes. Some devices allow multiple excitation current values such as 0.004 A and 0.01 A. Other devices allow only a single value such as 0.004 A. A value of 0 A disables the IEPE excitation. Refer to the device documentation for details on your device.

Note You cannot enable IEPE excitation on DSA devices when the terminal configuration is differential.

Note Changing the IEPE excitation level may cause transient voltages to appear in the signal. NI-DAQmx does not implement a delay to allow the signal to settle. Therefore, after changing the IEPE level and committing this change to the hardware with the Start function/VI or the Control Task function/VI, you might add a software delay to allow the signal to settle before proceeding with your application.

A DC offset is generated equal to the product of the excitation current and sensor impedance when IEPE signal conditioning is enabled. To remove the unwanted offset, you should enable AC coupling. Using DC coupling with IEPE excitation enabled is appropriate only if the offset does not exceed the voltage range of the channel.

Input Coupling

You can configure each AI channel of your DSA device to be either AC or DC coupled, with the exception of the NI 9233 and NI 449*X* devices, which are AC coupled only. If you select DC coupling, any DC offset present in the source signal is passed to the ADC. The DC-coupled configuration is usually best if the signal source has only small amounts of offset voltage, less than ±100 mV, or if the DC content of the acquired signal is important.

If the source has a significant amount of unwanted DC offset (bias voltage), you should select AC coupling to take full advantage of the input dynamic range. Selecting AC coupling enables a single-pole, high-pass resistor-capacitor (RC) filter into the positive and negative signal path. Refer to your device documentation for additional information on the filter circuitry.

Use the NI-DAQmx Channel attribute/property **AI.Coupling** to set the input coupling mode on your DSA device.

If you create a virtual channel for acceleration or sound pressure measurements with your DSA device, the input coupling for the channel defaults to AC. For other types of virtual channels, the input coupling defaults to DC.

Note NI-DAQmx does not compensate for the settling time or delay introduced by the RC filter.

Using AC coupling results in a drop in the low-frequency response of the AI circuitry. The AC coupling circuitry is usually characterized by a 3 dB cut-off frequency. However, the roll off from the high-pass filter can have a measurable effect even at frequencies several times greater than the 3 dB point.

Open Current Loop Detection

Two NI-DAQmx Read attributes/properties allow you to check for disconnected sensors. The first is **Open Current Loop Chans Exist**. This attribute/property returns a Boolean true if one or more channels experience an open current loop condition. The second is **Open Current Loop Chans**. This attribute/property returns an array of strings indicating which channels (if any) experienced an open current loop condition. You must query the **Open Current Loop Chans Exist** attribute/property before querying the **Open Current Loop Chans** attribute/property.

Open Current Loop Chans Exist reads the open current loop condition from the device and caches it in the driver. Subsequent reads of **Open Current Loop Chans** attribute/property will read the open current loop channel information cached in the driver from the previous **Open Current Loop Chans Exist** query.

Note NI-DAQmx returns all data whether or not an open current loop occurs. If your application requires open current loop checking, it is recommended that you read the open current loop attributes/properties after each call to Read. Your program should discard questionable data or return a flag when the driver reports an open current loop.

Note IEPE must be turned on for open current loop detection to work. If IEPE is not turned on, an error is returned when **Open Current Loop Chans** is read.

Overcurrent Detection

Two NI-DAQmx Read attributes/properties allow you to check for shorted channels. The first is **Overcurrent Chans Exist**. This attribute/property returns a Boolean true if one or more channels experience an overcurrent condition. The second is **Overcurrent Chans**. This attribute/property returns an array of strings indicating which channels (if any) experienced an overcurrent condition. You must query the **Overcurrent Chans Exist** attribute/property before querying the **Overcurrent Chans** attribute/property.

Overcurrent Chans Exist reads the overcurrent condition from the device and caches it in the driver. Subsequent reads of **Overcurrent Chans** attribute/property will read the overcurrent channel information cached in the driver from the previous **Overcurrent Chans Exist** query.

Note NI-DAQmx returns all data whether or not a short occurs. If your application requires overcurrent checking, it is recommended that you read the overcurrent attributes/properties after each call to Read. Your program should discard questionable data or return a flag when the driver reports a short.

Note IEPE must be turned on for overcurrent detection to work. If IEPE is not turned on, an error is returned when **Overcurrent Chans Exist** is read.

Overload Detection

DSA devices support overload detection in both the analog domain (pre-digitization) and digital domain (post-digitization). An analog overrange can occur independently from a digital overrange, and vice versa. For example, an IEPE accelerometer might have a resonant frequency that, when stimulated, can produce an overrange in the analog signal. However, because the delta-sigma technology of the ADC uses very sharp anti-aliasing filters, the overrange is not passed into the digitized signal. Conversely, a sharp transient on the analog side might not overrange, but the step response of the delta-sigma anti-aliasing filters might result in clipping in the digital data.

Some DSA devices support both analog and digital overload detection, while others support only digital overload detection. Consult your device documentation for more information on the overload detection capabilities for your device.

Two NI-DAQmx Read attributes/properties allow you to check for overloaded channels. The first is **OverloadChansExist**. This attribute/property returns a Boolean true if one or more channels experience an overload condition. The second is **OverloadedChans**. This attribute/property returns an array of strings indicating which channels (if any) experienced an overload condition. You must query the **OverloadChansExist** attribute/property before querying the **OverloadedChans** attribute/property.

OverloadChansExist reads the overload condition from the device and caches it in the driver. It also resets the overload condition of the device after it is read. Subsequent reads of **OverloadedChans** attribute/property will read the overloaded channel information cached in the driver from the previous **OverloadChansExist** query.

Note NI-DAQmx returns all data whether or not an overload occurs. If your application requires overload checking, it is recommended that you read the overload attributes/properties after each call to Read. Your program should discard questionable data or return a flag when the driver reports an overload.

Filter Delay

The delta-sigma ADCs and DACs on DSA devices and the NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239 employ digital filtering that imparts a delay of several sample intervals. The filter delays are equal in a homogeneous system, so these delays cancel out when performing phase measurements between channels. However, the filter delays differ in a heterogeneous system. This can introduce errors in phase comparisons between channels on different devices or between channels on similar devices running at different rates. Such phase errors are always deterministic, and you can account for them in software. Refer to your device documentation for more information on the digital filters and the delays associated with them.

DSA, C Series, and the DAQmx I/O Server

DSA devices and the NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239 do not support the DAQmx I/O Server.

Multidevice Tasks

Tasks can contain channels from multiple devices for these devices.

- C Series
- DSA
- S Series

C Series Multidevice Tasks

A task can include channels from multiple C Series devices, given the following conditions. When you include channels from multiple C Series devices in a task, NI-DAQmx automatically synchronizes the devices.

- All channels in the task must be of the same I/O type, but they cannot be counter I/O channels
- If the task includes channels from a mixture of NI 9225, NI 9229, NI 9233, NI 9234, NI 9235, NI 9236, NI 9237, and NI 9239 devices, you must account for filter delay differences between the devices.
- The devices must all be in the same NI cDAQ-9172 chassis.
- AI tasks containing only 16-bit or lower resolution AI modules use half the USB bandwidth of tasks with 24-bit AI modules.
- The format of raw data returned by a C Series AI task varies depending on if any 24-bit AI modules are in the task and might not correspond to the order of the channels in the task. Scaled or unscaled data is preferable to raw data with the NI cDAQ-9172 chassis.

DSA

A task can include channels from multiple DSA devices, given the following conditions. When you include channels from multiple DSA devices in a task, NI-DAQmx automatically synchronizes the devices.

All Devices

- All channels in the task must be analog input channels.
- If the task includes channels from both 446*X* and 447*X* devices, you must account for filter delay differences between the devices.

PXI Devices

- The devices must all be in a single chassis, and you must identify the chassis in MAX.
- If the task includes channels from 447*X* devices, one of the devices must be in PXI slot 2.
- If the task includes channels from both 446*X* and 447*X* devices, a 446*X* device must be in PXI slot 2.

PCI Devices

You must use a RTSI cable to connect the devices, and you must identify the cable in MAX.

S Series Multidevice Tasks

A task can include channels from multiple S Series devices, given the following conditions. When you include channels from multiple S Series devices in a task, NI-DAQmx automatically synchronizes the devices.

All Devices

All channels in the task must be analog input channels.

When you include channels from different S Series devices and use an external clock setup, you must import the external clock into the device with the longest pipeline of all the devices in the task. Failing to do so results in an incomplete acquisition, with the device importing the clock not receiving enough sample clock pulses.

PXI Devices

The devices must all be in a single chassis, and you must identify the chassis in MAX.

PCI Devices

You must use a RTSI cable to connect the devices, and you must identify the cable in MAX.

Initialized States for Terminals and Output Channels

When you use MAX or the NI-DAQmx API to reset a device, NI-DAQmx sets terminals and output channels to an initialized state.

Digital I/O Lines

NI-DAQmx sets all digital I/O lines to the configured power-up state. NI-DAQmx tristates all digital I/O lines on devices that do not support configurable power-up states. NI-DAQmx outputs 0 on all digital output-only lines on devices that do not support configurable power-up states.

PFI Lines

NI-DAQmx tristates all PFI lines, unless they are also digital I/O lines. In that case, the digital I/O line behavior applies.

AO Channels

On E Series, S Series, and AO Series devices, NI-DAQmx does not alter the AO channels. They continue to generate the DC voltage you last set them to.

With static AO devices, all voltage outputs are at their user-defined values to full accuracy within 1 s of power-up device reset. Before this time, the voltage outputs can float to unspecified values.

On DSA devices, NI-DAQmx sets all AO channels to high impedance.

On M Series and C Series, the AO channels are set to 0 Volts.

External Reference Sources

For 625*X* M Series devices, you can use APFI 0 or APFI 1 for analog output external reference sources. For 628*X* M Series devices, you can use ao0 through ao3. On some other STC-based devices, you can use EXTREF as the analog output external reference source. Using an external voltage reference enables you to maximize the resolution of your device. If the voltages you want to generate do not exceed a certain level and you can supply an external reference voltage at that level, you achieve your device's maximum resolution. You also can use external reference voltages to apply a gain to a DC voltage or to a time-varying waveform.

Refer to the specifications for your device for additional information.

Setting Power-Up States for M Series, NI 670*X*, and Software-Timed Digital I/O Devices

You can set the state of physical channels for some devices when your computer is powered on or the device is reset in NI-DAQmx. However, for all NI-DAQmx simulated devices, power-up states are not persisted.

Caution Devices have limited numbers of writes to the EEPROM, so change power-up states infrequently.

Setting Digital States for M Series and Software-Timed Digital I/O Devices

You can set the digital power-up state for M Series and software-timed digital I/O devices to logic low, logic high, or tristate (floating) in MAX. You also can set power-up states with the Set Power Up States (Digital) function/VI, but using MAX is the recommended method. You can only specify a programmable power-up state of tristate on devices with configurable direction. Refer to your device documentation to see if your device supports configurable direction. For NI 6230/36 devices, you can also specify a power-up state of tristate for digital output. The power-up state can be specified on a port basis only. The power-up state of all other NI 623*X* devices can be specified by line but cannot be set to tristate.

Note I/O direction on software-timed digital I/O devices is port configurable only. Therefore, you can set the power-up state to tristate only on a port-by-port basis. You can, however, set individual digital output lines in a port to logic low or logic high.

Setting Analog States for NI 670*X* Devices

You can set the analog power-up state for NI 670*X* devices in MAX. You also can set power-up states with the Set Power Up States (Analog) function/VI, but using MAX is the recommended method.

Push-Pull and Open Collector Mode

The NI USB-6008 has open-collector (open-drain) mode only, but each channel on the NI USB-6501, the NI USB-6009, and SensorDAQ are programmable as either push-pull or open-collector (open-drain) mode. The default configuration of the SensorDAQ, the NI USB-6008, the NI USB-6009, and the NI USB-6501 DIO ports is open-drain, allowing 5 V operation, with an onboard 4.7 k pull-up resistor. An external, user-provided, pull-up resistor can be added to increase the source current drive up to a 8.5 mA limit per line. Refer to the device documentation for more information and instructions on determining the value of the pull-up resistor.

Querying Device Capabilities with C Series Devices

When querying DAQmx Device and DAQmx Physical Channel attributes/properties with C Series devices, the supported attributes/properties depend on the slot you plug a device into. For instance, to use the counter functionality on a C Series device, you must plug the C Series device into slots 5 or 6. If the device is not in one of these slots, you cannot perform counter I/O, nor can you query counter-specific device capabilities such as the counter size.

RTSI Triggering with M Series USB Devices

M Series USB devices do not support RTSI triggering.

SCC Signal Conditioning Device Considerations

The following section applies only to analog input (AI) SCC modules.

In a single stage AI SCC configuration, you connect your external signal to an SCC module which conditions the signal and passes it to the DAQ device. You can install single stage AI modules in sockets J1-J8 in the SC-2345 carrier.

Sometimes, you can cascade two AI SCC modules together on a single AI channel to form a dual-stage configuration. In this configuration, you connect the external signal to the first-stage AI module, which conditions the signal and passes it to the second-stage AI module. Then, the signal is passed to the DAQ device. First-stage of dual-stage AI modules can be located in sockets J9-J16. Second-stage of dual-stage AI modules can be located in sockets J1-J8.

An example of a dual-stage configuration is a voltage attenuator module (SCC-A10) in the first-stage SCC slot followed by a lowpass filter module (SCC-LP01) in the second-stage SCC slot to create a combined attenuator and lowpass filter signal conditioning on the specified AI channel. The following table shows all the SCC devices that support dual-stage configuration.

SCC Module	Single Stage AI (J1-J8)	First-Stage of Dual-Stage AI (J9-J16)	Second-Stage of Dual-Stage AI (J1-J8)
SCC-AI Series	Yes	Yes	No
SCC-A10	Yes	Yes	No
SCC-RTD01	Yes	Yes	No
SCC-CI20	Yes	Yes	No
SCC-ACC01	Yes	Yes	No
SCC-TC Series	Yes	Yes	No
SCC-	Yes	No	Yes

FV01			
SCC-LP Series	Yes	Yes	Yes
SCC-FT01	Yes	Yes	Yes
SCC-SG Series	Yes	Yes	No

SCC modules that are not listed in the table above do not support dual-stage configuration. This includes all analog output, digital input, and digital output SCC modules.

Supported Device ID Numbers

The following table lists the device names and ID numbers for all devices NI-DAQmx supports. Click the column headings to sort the table.

Device ID	Device Name
0x0160	NI PCI-DIO-96
0x0162	NI PCI-MIO-16XE-50
0x0245	NI DAQCard-6036E
0x02C4	NI DAQCard-6062E
0x0301	NI SCXI-1126
0x0302	NI SCXI-1121
0x0303	NI SCXI-1129
0x0304	NI SCXI-1120
0x0306	NI SCXI-1100
0x0308	NI SCXI-1140
0x030A	NI SCXI-1122
0x030C	NI SCXI-1160
0x030D	NI SCXI-1125
0x030E	NI SCXI-1161
0x0310	NI SCXI-1162
0x0312	NI SCXI-1163
0x0314	NI SCXI-1124
0x0318	NI SCXI-1162HV
0x031C	NI SCXI-1163R
0x031E	NI SCXI-1102
0x031F	NI SCXI-1102B
0x0320	NI SCXI-1141
0x0321	NI SCXI-1112
0x0322	NI SCXI-1191
0x0323	NI SCXI-1190

0x0324	NI SCXI-1192
0x0329	NI SCXI-1600
0x032D	NI SCXI-1104
0x032F	NI SCXI-1104C
0x0330	NI SCXI-1531
0x0331	NI SCXI-1540
0x0332	NI SCXI-1530
0x033E	NI SCXI-1102C
0x0340	NI SCXI-1143
0x0344	NI SCXI-1120D
0x0345	NI SCXI-1127
0x0346	NI SCXI-1128
0x0360	NI SCXI-1142
0x038F	NI SCXI-1500
0x0407	NI SCXI-1520
0x048B	NI SCXI-1581
0x04AC	NI SCC-A10
0x04AD	NI SCC-CI20
0x04AE	NI SCC-FT01
0x04B2	NI SCC-TC01
0x04B3	NI SCC-TC02
0x04B4	NI SCC-FV01
0x04B6	NI SCC-DI01
0x04B7	NI SCC-DO01
0x04B8	NI SCC-LP01
0x04B9	NI SCC-LP02
0x04BA	NI SCC-LP03
0x04BC	NI SCC-LP04
0x04BD	NI SCC-RTD01
0x04BE	NI SCC-SG01

0x04BF	NI SCC-CO20
0x04DA	NI SCC-AI01
0x04DB	NI SCC-AI02
0x04DC	NI SCC-AI03
0x04DD	NI SCC-AI04
0x04DE	NI SCC-AI05
0x04DF	NI SCC-AI06
0x04E0	NI SCC-AI07
0x04E1	NI SCC-AI13
0x04E2	NI SCC-AI14
0x04E3	NI SCC-SG02
0x04E4	NI SCC-SG03
0x04E5	NI SCC-SG04
0x04E6	NI SCC-SG11
0x04EA	NI SCC-ACC01
0x04F0	NI SCC-RLY01
0x04F6	NI SCC-AO10
0x075C	NI DAQCard-DIO-24
0x075E	NI DAQCard-6024E
0x075F	NI DAQCard-6715
0x1150	NI PCI-DIO-32HS
0x1170	NI PCI-MIO-16XE-10
0x1180	NI PCI-MIO-16E-1
0x1190	NI PCI-MIO-16E-4
0x11B0	NI PXI-6070E
0x11C0	NI PXI-6040E
0x11D0	NI PXI-6030E
0x1270	NI PCI-6032E
0x1290	NI PCI-6704
0x12B0	NI PCI-6534

0x1310	NI PCI-6602
0x1320	NI PXI-6533
0x1330	NI PCI-6031E
0x1340	NI PCI-6033E
0x1350	NI PCI-6071E
0x1360	NI PXI-6602
0x13C0	NI PXI-6508
0x1490	NI PXI-6534
0x14E0	NI PCI-6110
0x14F0	NI PCI-6111
0x1580	NI PXI-6031E
0x15B0	NI PXI-6071E
0x1710	NI PXI-6509
0x17D0	NI PCI-6503
0x1870	NI PCI-6713
0x1880	NI PCI-6711
0x18B0	NI PCI-6052E
0x18C0	NI PXI-6052E
0x1920	NI PXI-6704
0x1930	NI 6040E
0x1AD0	NI PCI-6133
0x1AE0	NI PXI-6133
0x1E30	NI PCI-6624
0x1E40	NI PXI-6624
0x2410	NI PCI-6733
0x2420	NI PXI-6733
0x2430	NI PCI-6731
0x24B0	NI PXI-4200
0x24F0	NI PXI-4472
0x2510	NI PCI-4472

0x2520	NI PCI-4474
0x27A0	NI PCI-6123
0x27B0	NI PXI-6123
0x2890	NI PCI-6036E
0x28A0	NI PXI-4461
0x28B0	NI PCI-6013
0x28C0	NI PCI-6014
0x2A60	NI PCI-6023E
0x2A70	NI PCI-6024E
0x2A80	NI PCI-6025E
0x2AB0	NI PXI-6025E
0x2B10	NI PXI-6527
0x2B20	NI PCI-6527
0x2B80	NI PXI-6713
0x2B90	NI PXI-6711
0x2C60	NI PCI-6601
0x2C80	NI PCI-6035E
0x2C90	NI PCI-6703
0x2CA0	NI PCI-6034E
0x2CC0	NI PXI-6608
0x2EC0	NI PXI-6115
0x2ED0	NI PCI-6115
0x2EE0	NI PXI-6120
0x2EF0	NI PCI-6120
0x7023	NI PXI-2593
0x7024	NI SCXI-1193
0x703D	NI SCXI-1166
0x703E	NI SCXI-1167
0x703F	NI PXI-2566
0x7040	NI PXI-2567

0x704B	NI SCXI-1130
0x704C	NI PXI-2530
0x704F	NI PXI-4220
0x7050	NI PXI-4204
0x7067	NI PXI-2529
0x7073	NI PCI-6723
0x707E	NI PXI-4462
0x7085	NI PCI-6509
0x7086	NI PXI-6528
0x7087	NI PCI-6515
0x7088	NI PCI-6514
0x708C	NI PXI-2568
0x708D	NI PXI-2569
0x7090	NI SCXI-1169
0x7099	NI SCC-SG24
0x709F	NI USB-9421
0x70A1	NI USB-9472
0x70A2	NI USB-9481
0x70A4	NI USB-9201
0x70A5	NI USB-9221
0x70A7	NI USB-9263
0x70A8	NI USB-9233
0x70A9	NI PCI-6528
0x70AA	NI PCI-6229
0x70AB	NI PCI-6259
0x70AC	NI PCI-6289
0x70AD	NI PXI-6251
0x70AE	NI PXI-6220
0x70AF	NI PCI-6221
0x70B0	NI PCI-6220

0x70B1	NI PXI-6229
0x70B2	NI PXI-6259
0x70B3	NI PXI-6289
0x70B4	NI PCI-6250
0x70B5	NI PXI-6221
0x70B6	NI PCI-6280
0x70B7	NI PCI-6254
0x70B8	NI PCI-6251
0x70B9	NI PXI-6250
0x70BA	NI PXI-6254
0x70BB	NI PXI-6280
0x70BC	NI PCI-6284
0x70BD	NI PCI-6281
0x70BE	NI PXI-6284
0x70BF	NI PXI-6281
0x70C0	NI PCI-6143
0x70C3	NI PCI-6511
0x70C8	NI PCI-6513
0x70C9	NI PXI-6515
0x70CC	NI PCI-6512
0x70CD	NI PXI-6514
0x70D0	NI PXI-2570
0x70D1	NI PXI-6513
0x70D2	NI PXI-6512
0x70D3	NI PXI-6511
0x70D4	NI PCI-6722
0x70E1	NI PXI-2532
0x70F2	NI PCI-6224
0x70F3	NI PXI-6224
0x70FA	NI DAQPad-6015

0x70FB	NI DAQPad-6016
0x70FF	NI PXI-6723
0x7100	NI PXI-6722
0x7103	NI SCXI-1521
0x710D	NI PXI-6143
0x7124	NI PCI-6510
0x7125	NI PCI-6516
0x7126	NI PCI-6517
0x7127	NI PCI-6518
0x7128	NI PCI-6519
0x712E	NI USB-9421 (DSUB)
0x7132	NI USB-9472 (DSUB)
0x7137	NI PXI-2575
0x713C	NI PXI-2585
0x713D	NI PXI-2586
0x713F	NI SCXI-1175
0x7142	NI PXI-4224
0x7143	NI SCXI-1521B
0x7146	NI PCI-6132
0x7147	NI PXI-6132
0x7148	NI PCI-6122
0x7149	NI PXI-6122
0x7150	NI PXI-2564
0x715F	NI 9221
0x7160	NI 9421
0x7161	NI 9421 (DSUB)
0x7162	NI 9472
0x7163	NI 9472 (DSUB)
0x7164	NI 9481
0x7165	NI 9401

0x716B	NI PCI-6230
0x716C	NI PCI-6225
0x716D	NI PXI-6225
0x716F	NI PCI-4461
0x7170	NI PCI-4462
0x7171	NI PCI-6010
0x7172	NI DAQPad-6015 (Mass Termination)
0x7173	NI DAQPad-6015 (BNC)
0x7177	NI PXI-6230
0x717A	NI USB-6008
0x717B	NI USB-6009
0x717D	NI PCIe-6251
0x717F	NI PCIe-6259
0x718A	NI USB-6501
0x718B	NI PCI-6521
0x718C	NI PXI-6521
0x7191	NI PCI-6154
0x71A1	NI USB-9201 (DSUB)
0x71A2	NI USB-9221 (DSUB)
0x71A5	NI PXI-2594
0x71A6	NI SCXI-1194
0x71A7	NI PXI-2595
0x71A8	NI SCXI-1195
0x71A9	NI PXI-2596
0x71AA	NI PXI-2597
0x71AB	NI PXI-2598
0x71AC	NI PXI-2599
0x71B0	NI 9211
0x71B1	NI 9215
0x71B2	NI 9215 (BNC)

0x71B3	NI 9205 (DSUB)
0x71B4	NI 9263
0x71BB	NI PXI-2584
0x71BC	NI PCI-6221 (37-pin)
0x71C2	NI USB-9239
0x71C3	NI USB-9237
0x71C5	NI PCI-6520
0x71C6	NI PXI-2576
0x71D9	NI USB-9211A
0x71DA	NI USB-9215A
0x71DB	NI USB-9215A (BNC)
0x71DF	NI USB-6525
0x71E0	NI PCI-6255
0x71E1	NI PXI-6255
0x71E7	NI 9233
0x71E8	NI SCXI-1502
0x7209	NI PCI-6233
0x720A	NI PXI-6233
0x720B	NI PCI-6238
0x720C	NI PXI-6238
0x7252	NI USB-6251
0x7253	NI USB-6259
0x7263	NI 9234
0x7264	NI 9206
0x7265	NI 9205
0x726A	NI SCXI-1503
0x726E	NI SCC-CTR01
0x726F	NI USB-6210
0x7270	NI USB-6211
0x7271	NI USB-6215

0x7272	NI USB-6218
0x7273	NI PXI-4461
0x7274	NI PXI-4462
0x7279	NI PCI-6232
0x727A	NI PXI-6232
0x727B	NI PCI-6239
0x727C	NI PXI-6239
0x7281	NI PCI-6236
0x7282	NI PXI-6236
0x7283	NI PXI-2554
0x7285	NI 9237
0x72A0	NI USB-6251 (Mass Termination)
0x72A1	NI USB-6259 (Mass Termination)
0x72B5	NI USB-9234
0x72B9	NI 9411
0x72BA	NI 9422
0x72BB	NI 9423
0x72BC	NI 9435
0x72BD	NI 9474
0x72BE	NI 9485
0x72BF	NI 9403
0x72C0	NI 9425
0x72C1	NI 9476
0x72C2	NI 9477
0x72C3	NI 9264
0x72C4	NI 9265
0x72C5	NI 9201
0x72C6	NI 9201 (DSUB)
0x72C7	NI 9221 (DSUB)
0x72C8	NI 9203

0x72C9	NI 9217
0x72CA	NI 9219
0x72CB	NI 9239
0x72CC	NI SensorDAQ
0x72D0	NI PXI-2545
0x72D1	NI PXI-2546
0x72D2	NI PXI-2547
0x72D3	NI PXI-2548
0x72D4	NI PXI-2549
0x72D5	NI PXI-2555
0x72D6	NI PXI-2556
0x72D7	NI PXI-2557
0x72D8	NI PXI-2558
0x72D9	NI PXI-2559
0x72DC	NI USB-6221
0x72DE	NI USB-6229
0x72EF	NI PXI-4498
0x72F0	NI PXI-4496
0x72F3	NI USB-6005 VSA
0x72FA	NI 9229
0x72FD	NI USB-9229
0x72FF	NI USB-6509
0x730C	NI USB-9219
0x731C	NI PXI-2535
0x731D	NI PXI-2536
0x7327	NI PXI-6529
0x732D	NI USB-6255
0x732E	NI USB-6255 (Mass Termination)
0x732F	NI USB-6225
0x7330	NI USB-6225 (Mass Termination)

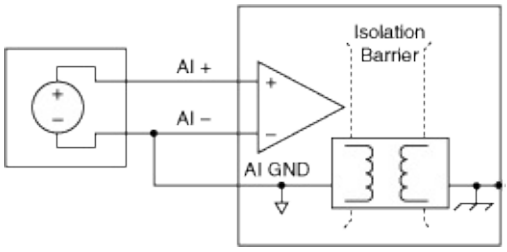
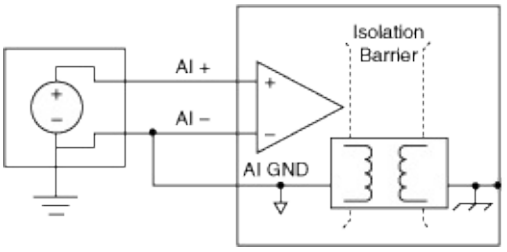
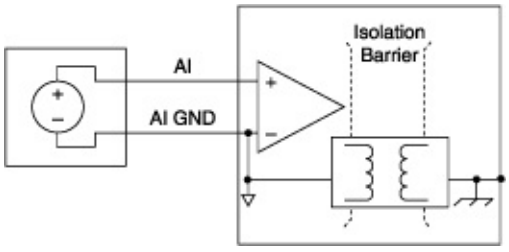
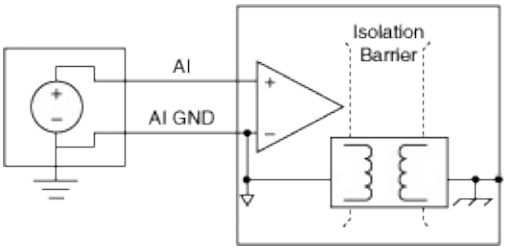
0x7335	NI PXI-2533
0x7336	NI PXI-2534
0x7337	NI 9402
0x7339	NI USB-6212
0x733B	NI USB-6216
0x733F	NI USB-6281
0x7340	NI USB-6281 (Mass Termination)
0x7342	NI PXI-4461
0x7343	NI USB-6289
0x7344	NI USB-6289 (Mass Termination)
0x7345	NI USB-6221 (BNC)
0x7346	NI USB-6229 (BNC)
0x7347	NI USB-6251 (BNC)
0x7348	NI USB-6259 (BNC)
0x7359	NI PXI-4495
0x7367	NI USB-9239 (BNC)
0x7368	NI USB-9229 (BNC)
0x7369	NI USB-9263 (BNC)
0x7370	NI PXI-4461
0x737A	NI 9229 (BNC)
0x737B	NI 9239 (BNC)
0x737C	NI 9263 (BNC)
0x7381	NI 9235
0x7382	NI 9236
0x7388	NI 9225
0x73C5	NI PXIe-2527
0x73C6	NI PXIe-2529
0x73C8	NI PXIe-2530
0x73C9	NI PXIe-2532
0x73CA	NI PXIe-2569

0x73CB	NI PXIe-2575
0x73CC	NI PXIe-2593
0x9020	NI PXI-2501
0x9030	NI PXI-2503
0x9040	NI PXI-2527
0x9050	NI PXI-2565
0x9060	NI PXI-2590
0x9070	NI PXI-2591

Connecting Analog Voltage Input Signals for Isolated Devices

Connecting Analog Voltage Input Signals

The following table summarizes the recommended input configuration for both types of signal sources:

Input	Signal Source Type	
	Floating Signal Sources (Not Connected to Building Ground)	**Ground-Referenced Signal Sources**
	Examples: • Ungrounded thermocouples • Signal conditioning with isolated outputs • Battery devices	Example: • Plug-in instruments with non-isolated outputs
Differential (DIFF)		
Referenced Single-Ended (RSE)		

Refer to Terminal Configurations (Analog Input Ground Reference Settings) for Isolated Devices for descriptions of the RSE and DIFF modes.

Self-Powered Versus Bus-Powered M Series USB Devices

Bus-powered M Series devices require no external power source. Examples of these devices include the NI 6210, NI 6211, NI 6212, NI 6215, NI 6216, and NI 6218. Self-powered M Series devices, on the other hand, require an external power source such as a wall outlet or a battery. Self-powered devices can drive higher currents than bus-powered devices. Examples of self-powered M Series devices include the NI USB-6251 and NI USB-6259.

Power Supply Fault Detection

The NI 9265 can detect if there is an insufficient external power or no external power connected to the Vsup terminal. Two NI-DAQmx Read attributes/properties allow you to check the external power. The first is **Power Supply Fault Chans Exist**. This attribute/property returns a Boolean true if one or more channels lack external power. The second is **Power Supply Fault Chans**. This attribute/property returns an array of strings indicating which channels (if any) experienced a power supply fault. You must query the **Power Supply Fault Chans Exist** attribute/property before querying the **Power Supply Fault Chans** attribute/property.