

---

Version 8.0

# **VisSim/Real-TimePRO User's Guide**

**By Visual Solutions, Inc.**

---

**Visual Solutions, Inc.**

**VisSim/Real-TimePRO User's Guide Version 8.0**

Copyright	© 2010 Visual Solutions, Inc. All rights reserved.	Visual Solutions, Inc. 487 Groton Road Westford, MA 01886
Trademarks	VisSim, VisSim/Analyze, VisSim/CAN, VisSim/C-Code, VisSim/C-Code Support Library source, VisSim/Comm, VisSim/Comm C-Code, VisSim/Comm Red Rapids, VisSim/Comm Turbo Codes, VisSim/Comm Wireless LAN, VisSim/Fixed-Point, VisSim/Knobs & Gauges, VisSim/Model-Wizard, VisSim/Motion, VisSim/Neural-Net, VisSim/OPC, VisSim/OptimizePRO, VisSim/Real-TimePRO, VisSim/State Charts, VisSim/Serial, VisSim/UDP, VisSim Viewer, and flexWires are trademarks of Visual Solutions. All other products mentioned in this manual are trademarks or registered trademarks of their respective manufacturers.	
Copyright and use restrictions	The information in this manual is subject to change without notice and does not represent a commitment by Visual Solutions. Visual Solutions does not assume any responsibility for errors that may appear in this document.  No part of this manual may be reprinted or reproduced or utilized in any form or by any electronic, mechanical, or other means without permission in writing from Visual Solutions. The Software may not be copied or reproduced in any form, except as stated in the terms of the Software License Agreement.	
Acknowledgements	The following engineers contributed significantly to preparation of this manual: Mike Borrello, Allan Corbeil, and Richard Kolk.	

---

# Contents

<b>Introduction</b>	<b>1</b>
The VisSim product family .....	1
Resources for learning VisSim/Real-TimePRO .....	3
Interactive webinars .....	3
Sample diagrams .....	3
Training .....	3
<b>The Basics</b>	<b>5</b>
Real-time drivers .....	5
Supported boards .....	6
The Real Time blocks .....	6
Data acquisition guidelines .....	6
Performance degradation .....	7
<b>Configuring ISA Bus I/O Boards</b>	<b>9</b>
Installing and configuring I/O boards .....	9
Finding out an I/O board's base address .....	9
Configuring VisSim/Real-TimePRO to recognize your I/O boards .....	9
Using the File > Real Time Config command .....	10
Setting the board type and number .....	10
Setting the analog input and output ranges .....	10
Setting the base address .....	11
Setting the channel settling time .....	11
Setting the external mux settling time .....	11
<b>Setting Input and Output Channels with the RT Blocks</b>	<b>13</b>
Input and output channel basics .....	13
Using the rt-DataIn block .....	13
Specifying the board number .....	14
Specifying a channel title .....	14
Specifying a channel number .....	14
Specifying subchannels .....	15
Specifying a mux gain for a multiplex board .....	15
Specifying a channel class .....	15
Specifying a channel type .....	17
Specifying cold junction .....	17
Using the rt-DataOut block .....	18
Specifying the board number .....	18
Specifying a channel title .....	18
Specifying a channel number .....	18
Specifying a channel class .....	19
Specifying a channel type .....	20

<b>Using the NI-DAQmx Blocks</b>	<b>23</b>
NI-DAQmx Analog In.....	23
Options .....	23
Parameters .....	24
Advanced .....	35
NI-DAQmx Digital In .....	36
NI-DAQmx Counter In.....	37
Options .....	38
Parameters .....	38
NI-DAQmx Analog Out .....	44
NI-DAQmx Digital Out.....	45
NI-DAQmx Counter Out .....	46
Reset command .....	47
<b>Using the Measurement Computing Universal Driver Blocks</b>	<b>49</b>
Using the MCC Input block .....	49
Using the MCC Output block.....	50
<b>Producing Pulse Width Modulated Output</b>	<b>53</b>
Boards supported by the PWM block.....	53
Using the PWM block .....	53
Specifying properties of the PWM block .....	54
<b>Measuring Frequency Input</b>	<b>55</b>
Boards supported by the frequency block .....	55
Using the frequency block.....	55
Specifying properties of a frequency block .....	55
Specifying a sampling rate .....	56
<b>Reading Encoder Input</b>	<b>57</b>
Boards supported by the encoder block.....	57
Using the encoder block .....	57
Specifying properties of an encoder block .....	57
Specifying a title .....	58
Specifying the base address.....	58
Specifying the counter reset value.....	58
Specifying the channel .....	58
Specifying the signal inversion .....	58
Specifying the counter set mode value.....	58
Determining the direction of rotation value.....	59
<b>Developing Servo Control Systems</b>	<b>61</b>
Boards supported by the Tech80 5650A block.....	61
Using a Tech80 5650A block .....	61
Tech80 5650A block input .....	61
Tech80 5650A block outputs.....	62
Configuring a Tech80 5650A block .....	62
Specifying a control type.....	63
Setting up a profile .....	63
Setting the zero position.....	64
Setting up the base address for each axis .....	64

Setting up motor resolution .....	64
Specifying an output type.....	64
<b>Connecting to Boards with an 8255 Chip</b>	<b>67</b>
Configured as all input channels .....	67
Configured as all output channels .....	67
Configured as 12 inputs and 12 outputs .....	68
<b>Connecting to PCM-CIA DAS 16/12 and DAS 16/16 Boards</b>	<b>69</b>
Configured as four inputs and four outputs .....	69
<b>Installing VisSim/Real-TimePRO</b>	<b>71</b>
Installation requirements .....	71
Installation procedure .....	71
<b>Index</b>	<b>73</b>



# Introduction

VisSim/Real-TimePRO extends capabilities by providing the ability to interface with popular analog and digital I/O boards and devices from National Instruments, Measurement Computing, and Quanser.

VisSim/Real-TimePRO provides the capability to couple a VisSim system model directly with a real PLC or DCS system for off-line tuning, or to prototype a VisSim control with a real plant.

With VisSim/Real-TimePRO, hardware-in-the-loop (HIL) systems can be configured and executed by interfacing VisSim plant or controller models with real-world hardware, such as manufacturing plants, chemical processes, motors, pumps, and electric drives. The interface to the real-world hardware is through computer I/O cards; high-speed motion control interface cards; or serial port connections to Programmable Logic Controllers (PLCs) or Distributed Control Systems (DCSs). There is no code generation or programming involved to configure an HIL system with VisSim/Real-TimePRO.

---

## The VisSim product family

The VisSim product family includes several base products and product suites, as well as a comprehensive set of targeted add-on modules that address specific problems in areas such as data communications, data acquisition, linearization and analysis, and digital signal processing.

### Base products and product suites

Product	Function
Professional VisSim	Model-based design, simulation, testing, and validation of dynamic systems.  A personal version, VisSim PE, is also available. VisSim PE limits diagram size to 100 blocks.
VisSim/Comm Suite	Simulates end-to-end communication systems at the signal level using 200+ communications, signal processing, and RF blocks.  Includes Professional VisSim and VisSim/Comm blockset.  A personal version, VisSim/Comm Suite PE, is also

	<p>available. VisSim/Comm PE limits diagram size to 100 blocks and limits the Communication blockset. See the VisSim/Comm datasheet for details.</p> <p>VisSim/Comm Suite add-on modules are available for real-time data acquisition (Red Rapids digital tuner card); modeling PCCC turbo codes, including UMTS specification; and for support of Bluetooth, 802.11 a/b/g (Wi-Fi), and ultrawideband wireless designs.</p>
VisSim/Embedded Controls Developer Suite	<p>Rapidly prototypes and creates embedded controls for DSPs, DSCs, and MSP430 microcontrollers. You can simulate and generate scaled, fixed-point ANSI C code, as well as code for on-chip peripherals.</p> <p>Includes Professional VisSim, VisSim/C-Code, VisSim/Fixed-Point, and one user-specified target support.</p> <p>A personal version, VisSim/Embedded Controls Developer PE, is also available. VisSim/Embedded Controls Developer PE limits diagram size to 100.</p>
VisSim Viewer (free)	Lets you share VisSim models with colleagues and clients not licensed to use VisSim.

### Add-on modules

Add-On Module	Function
VisSim/Analyze	Performs frequency domain analysis of a linearized nonlinear subsystem.
VisSim/CAN	Interfaces with a USB CAN device to read and write CAN messages on the CAN bus.
VisSim/C-Code	Generates highly-optimized, ANSI C code that can be compiled and run on any platform that supports an ANSI C compiler.
VisSim/C-Code Support Library Source	Provides source code for the Support Library.
VisSim/Comm blockset	<p>Simulates end-to-end communication systems at the signal level using 200+ communications, signal processing, and RF blocks.</p> <p>A personal version, VisSim/Comm PE, is also available. VisSim/Comm PE is a subset of the Communication blockset. See the VisSim/Comm datasheet for details</p> <p>You can purchase VisSim/Comm add-on modules for real-time data acquisition (Red Rapids digital tuner cards); for modeling PCCC turbo codes, including UMTS specification; for support of Bluetooth, 802.11 a/b/g (Wi-Fi), and ultrawideband wireless designs.</p>
VisSim/Fixed-Point	Simulates the behavior of fixed-point algorithms prior to code generation and implementation of the algorithm on the fixed-point target.
VisSim/Knobs and Gauges	Provides dynamic gauges, meters, and knobs for process control, and measurement and validation systems.
VisSim/Model-Wizard	Generates transfer function model from historic or real-

	time data.
VisSim/Motion	Simulates motor control systems with customizable amplifiers, controllers, filters, motors, sensors, sources, tools, and transforms.
VisSim/Neural-Networks	Performs nonlinear system identification, problem diagnosis, decision-making prediction, and other problems where pattern recognition is important.
VisSim/OPC	Connects to any OPC server and log data or run a virtual plant in VisSim for offline tuning.
VisSim/OptimizePRO	Performs generalized reduced gradient method of parameter optimization.
VisSim/Real-TimePRO	Performs real-time data acquisition and signal generation using I/O cards, PLCs, and DCSs.
VisSim/Serial	Performs serial I/O with other computers.
VisSim/State Charts	Creates, edits, and executes event-based systems.
VisSim/UDP	Performs data exchange over the internet using UDP.
VisSim Viewer (free)	Lets you share VisSim models with colleagues and clients not licensed to use VisSim.

---

## Resources for learning VisSim/Real-TimePRO

For those of you that are new to VisSim, we have provided several free services to make your transition to VisSim fast, smooth, and easy:

### Interactive webinars

Interactive webinars offer you the opportunity to meet with Visual Solutions product specialists who will introduce and demonstrate our software products live on your computer and answer any questions you have. Each webinar is approximately 45 minutes long. To learn more about our interactive webinars, go to <http://www.vissim.com/webinars/webinars.html>.

### Sample diagrams

VisSim 8.0 includes a directory of fully documented sample diagrams. These diagrams illustrate both simple and complex models spanning a broad range of engineering disciplines, including aerospace, biophysics, chemical engineering, control design, dynamic systems, electromechanical systems, environmental systems, HVAC, motion control, process control, and signal processing.

#### To access sample diagrams

Click on the **Diagrams** menu in VisSim.

Click on **Examples > Applications**.

### Training

Visual Solutions offers training sessions for learning and gaining expertise in VisSim and the VisSim family of add-on products. Training sessions are conducted at Visual Solutions training facility in Westford, MA, as well as at customer sites and as online webinars.

For information on setting up a training session, contact [sales@vissol.com](mailto:sales@vissol.com).

# The Basics

Simulating in real-time mode has the effect of retarding a simulation so that one simulation second equals one second in real time. Typically, you use real-time mode for data acquisition or hardware-in-the-loop control situations. For this, however, you need VisSim/Real-TimePRO.

---

## Real-time drivers

There are three sets of real-time drivers supplied with VisSim/Real-TimePRO:

- 32-bit ISA bus drivers for Windows NT/2000/XP and Win 9x that include:
  - A protected-mode VxD that is automatically installed (Windows 9x)
  - A Windows NT device driver `rtDevice.sys` (Windows NT/2000/XP)
  - A VisSim Dynamic Link Library (DLL) file that provides the link between VisSim and the low level driver.
  - All real-time drivers support up to sixteen boards simultaneously. You can mix different boards and drivers or use multiple instances of the same board. You set the base I/O port address for the board through the [File menu's Real Time Config command](#).
- Measurement Computing Universal driver that works with all ISA bus, PMCIA, PCI and cPCI boards from Measurement Computing.
  - You must install Instacal first. It comes on the CD you receive with your Measurement Computing boards
- National Instruments NIDAQ driver. This driver supports all ISA, PCI, PXI, and PMCIA boards from National Instruments.
  - You must first install NIDAQ before installing this driver.

If you have a PC-based board, you must install either the Measurement Computing or National Instruments driver.

All three drivers are included in the VisSim/Real-TimePRO installation. However, you must enable the Measurement Computing or National Instruments option during the installation if you want to have them available in VisSim.

---

## Supported boards

For the most current list of boards supported by VisSim/Real-TimePRO, refer to [http://www.vissim.com/support/rt\\_boards.html](http://www.vissim.com/support/rt_boards.html).

---

## The Real Time blocks

VisSim/Real-TimePRO offer the following blocks:

Block	What it's for
rt-DataIn	Configuring input channels
rt-DataOut	Configuring output channels
MCC Input	Configuring input channels for Measurement Computing boards using universal driver
MCC Output	Configuring output channels for Measurement Computing boards using universal driver
PWM	Producing pulse width modulated output
frequency	Measuring frequency input
encoder	Reading encoder input

These blocks are located under the Real Time category in the Blocks menu. After VisSim/Real-TimePRO has been successfully installed, the `rt-DataIn` and `rt-DataOut` blocks are available for use. The `MCC Input`, `MCC Output`, `PWM`, `frequency`, `encoder`, and `Tech80 5650A` blocks are available only if you have installed the following boards:

To use this block	One of the following boards must be installed
MCC Input	<i>Measurement Computing</i> : any Measurement Computing board
MCC Output	<i>Measurement Computing</i> : any Measurement Computing board
PWM	<i>Analog Devices</i> : RTI 815; <i>Measurement Computing</i> : CIO-CTR05, CIO-CTR05/H50, CIO-CTR10, CIO-CTR10/H50
frequency	<i>Analog Devices</i> : RTI 815; <i>Measurement Computing</i> : CIO-CTR05, CIO-CTR05/H50, CIO-CTR10, CIO-CTR10/H50
encoder	<i>Precision Micro Dynamics</i> : MFIO-3A
Tech80 5650A	<i>Technology 80</i> : 5650A

---

## Data acquisition guidelines

Since Windows is not a multitasking operating system you must be careful about what you do while acquiring data to avoid missing a sampling interval. Because VisSim calculates all blocks before it begins each sampling interval, any one block that runs slowly can disturb the acquisition process. Additionally, since there is only one processor to calculate and draw the Windows graphics, you should avoid any operation that causes Windows to repaint the screen. These operations include:

- Running DOS shells

- Digital display read-outs, strip charts, histograms, meters, and plots for greater than 200 Hz sampling rates (shown in order, from most costly to least costly)
- Calling up a block's Properties dialog box
- Expanding a plot to full screen
- Changing diagram layers

---

## Performance degradation

The performance degradation due to graphics use depends heavily on the speed of the base processor and the graphics card. You should experiment with different sampling rates and graphics usage to determine the limits of your system.



# Configuring ISA Bus I/O Boards

This section contains...

---

## Installing and configuring I/O boards

Typically, the installation and configuration of your I/O boards involves setting base addresses, adjusting jumpers, and setting analog and digital ranges. Because this information is unique to each board, it is important that you refer to the vendor's documentation for installing, configuring, and testing the board.

### Keeping a record of your I/O board base address

You will need your I/O board's base address later on when you configure VisSim/Real-TimePRO to recognize the board. Therefore, it is recommended that you write down your boards' base addresses and keep them in a safe place.

---

## Finding out an I/O board's base address

If you need to determine the base address of one of your installed I/O boards, refer to the vendor's documentation. In many cases, the vendor distributes software that allows you to check installed board settings.

Also, you may be able to use the Control Panel to determine base addresses for installed boards. For instance, on the Windows 95 platform, you can find many boards' settings by using the Device Manager under Settings > Control Panel > Device Manager. Refer to your Windows documentation for information on finding I/O board settings.

---

## Configuring VisSim/Real-TimePRO to recognize your I/O boards

Any time you install or reconfigure an I/O board, you must also configure or reconfigure VisSim/Real-TimePRO to recognize the board. You use the File > Real-Time Config command to perform these tasks.

### When using the Measurement Computing universal driver

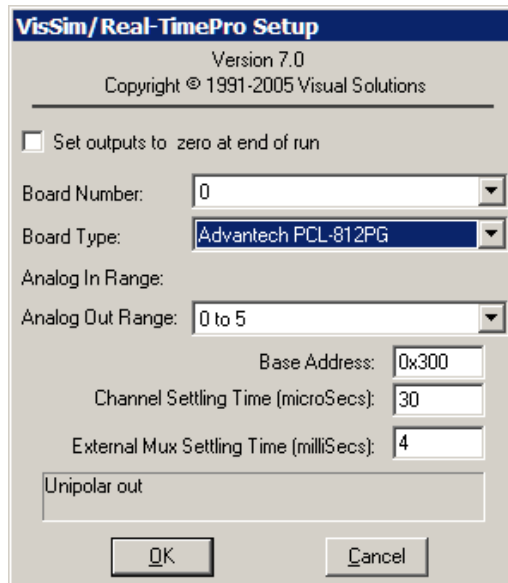
If you are using Measurement Computing's universal driver, use the INSTACAL program to configure your I/O boards. INSTACAL is on the CD that comes with your board.

## Using the File > Real Time Config command

After installing your I/O board, follow these steps to configure VisSim/Real-TimePRO to recognize it:

1. Start VisSim.
2. Choose **File > Real Time Config**.

A setup dialog box appears.



3. Select the hardware and software settings for the I/O board that will work with the process or controller whose data you are acquiring or measuring. (See the descriptions below for more information about each parameter.)
4. Click on the **OK** button, or press **ENTER**.

### Setting the board type and number

The Board Number box lets you identify your I/O board with a unique number in the range of 0 through 15. For instance, to differentiate two installed I/O boards, you can assign one board number 0 and the other board number 1. The number you enter here is used to identify your board later on when setting its input and output channels.

The Board Type box identifies the model of your I/O board. When you click in the box or on the DOWN ARROW, a list of available board types is displayed from which you can choose.

### Setting the analog input and output ranges

The Analog Input Range box lists the available analog input ranges for the selected board type, while the Analog Output Range box lists the available analog output

ranges for the selected board type. Click on the DOWN ARROW to reveal a list of additional ranges.

## Setting the base address

The base address indicates the I/O port register address through which the real-time driver commands the board. The base address was set at the time the board was installed and configured. Enter the base address as a hexadecimal number in the following format:

*0xhexNumber*

Base addresses are configurable between 0x110 and 0x3FF.

## Setting the channel settling time

The channel settling time is the time taken for the voltage to settle between channels for multichannel reads. The settling time is entered in microseconds ( $\mu\text{sec}$ ). The first time you configure a board, it is recommended to use the default setting; you can always change the setting later if it is too fast or too slow.

Most boards have a default settling time of 30  $\mu\text{sec}$ .

## Setting the external mux settling time

The external mux settling time is the time taken for the voltages to settle when using the I/O boards with multiplexed subchannels. The default is 4 milliseconds (msec).



# Setting Input and Output Channels with the RT Blocks

This section contains...

---

## Input and output channel basics

The `rt-DataIn` and `rt-DataOut` blocks let you set the input and output channels for your I/O boards. Each block sets a single channel. Thus, if you are using a 4 input, 4 output I/O board, you will need four `rt-DataIn` blocks and four `rt-DataOut` blocks.

For each channel, you can specify a channel title, channel class, and channel type. In addition, if you are setting the input channel on an I/O board with multiplexed subchannels, you can also set the multiplex subchannel, mux gain, and cold junction compensation for thermocouple linearization.

---

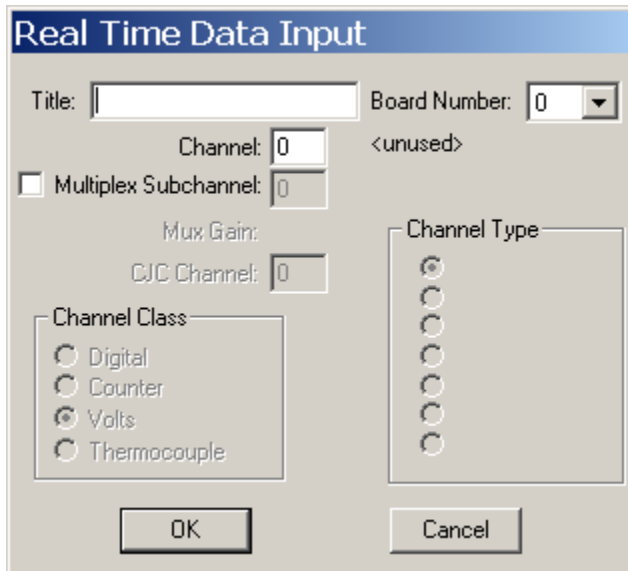
## Using the `rt-DataIn` block

To set an input channel on your I/O board, you use the `rt-DataIn` block.

To set up an input channel

1. Choose **Blocks > Real Time** and drag an **rt-DataIn** block into the work area.
2. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the **rt-DataIn** block.
  - Click the right mouse button over the **rt-DataIn** block.

The Real Time Data Input dialog box appears.



3. Make the appropriate selections. (See the descriptions below for more information about each parameter.)
4. Click on the **OK** button, or press **ENTER**.

## Specifying the board number

In the Board Number box, enter the board number of the I/O board whose input channel is to be configured. The board number was established with the [File > Real Time Config command](#). If you are not sure of the board number, click on the DOWN ARROW to reveal the list of configured board numbers. When you select a board number, the board type is displayed beneath the board number.

## Specifying a channel title

When you configure a multi-channel I/O board, you can identify each channel with a unique title. These titles appear on the rt-DataIn blocks making it easier to recognize which block is assigned to each channel.

## Specifying a channel number

In the Channel box, you enter a number that corresponds with the channel number on the screw terminal or termination panel supplied with your I/O board. The rt-DataIn block uses channel 0 as the first channel, even if the documentation supplied by the board vendor describes the first channel as channel 1.

[8255-based digital I/O boards](#): If you are using an 8255-based digital I/O board, you can connect rt-DataIn channels to electrical connector terminals in several different ways. Boards with an 8255 chip are listed below:

Vendor	Board
Measurement Computing	CIO-DIO24
	CIO-DIO24H
Keithley MetraByte	PIO-12

	PIO-24
National Instruments	PC-DIO-24
	PC-DIO-96
	DAQCard-1200

[PCM-CIA DAS 16/12 and DAS 16/16 boards](#): If you are connecting to a Measurement Computing PCM-CIA DAS 16/12 or DAS 16/16, you can connect the rt-DataIn channels to electrical terminals in a specific way.

## Specifying subchannels

When you are using an I/O board with multiplexed input channels, such as the CIO-DAS08 board from Measurement Computing, you use the Multiplex Subchannel parameter to connect the individual multiplex channels. Each multiplex board physically connects to all of the analog input channels on the I/O board, letting you daisy chain them. Electrically, each multiplex board only connects to one of the I/O board's input channels. The electrical connection connects to only one of the analog input channels and depends on the jumper setting on the multiplex board. To set the jumper, refer to the documentation that accompanies the board.

When you activate the Multiplex Subchannel parameter, you must also enter a subchannel number in the accompanying Multiplex Subchannel text box. VisSim/Real-TimePRO sends the multiplex subchannel number to the first four digital output channels, which the multiplex board uses as the multiplex subchannel specifier, and then reads the input channel. (You should not use the digital channels for other purposes.)

## Specifying a mux gain for a multiplex board

The Mux Gain parameter indicates the gain applied to a multiplexed signal when it is read into VisSim/Real-TimePRO. In order to render an accurate reading, VisSim/Real-TimePRO uses this parameter to compensate for the gain applied to the signal by the multiplexor board.

The Mux Gain parameter value must match the gain setting specified on the multiplexor board to achieve accurate results.

When you change the mux gain on a channel, all the gains on all the multiplexed subchannels are changed.

## Specifying a channel class

The rt-DataIn block recognizes six classes of inputs on I/O boards:

- Counter
- Current
- Digital
- Thermocouple
- Voltage

When you select a channel class, the options listed under the Channel Type box change accordingly. For example, when you select Volts, the voltage ranges are displayed.

In addition, before you select a channel class, make sure the hardware line has been connected to the specified channel number.

### **Counter channel class**

The Counter parameter utilizes your I/O board's counter timer. A counter provides a high frequency pulse counting input. Pulses are generated by triggering on the leading edge of an incoming sinusoidally-shaped waveform. Most counters can count approximately 64,000 pulses before they overflow and reset. In VisSim/Real-TimePRO, the counter value is reset at the end of every time step of the simulation. If you accumulate more than 64,000 counts, you should reduce the step size using the Simulate > Simulation Properties command.

Most boards have at least one counter input. When using the counter channel, be particularly careful connecting the screw terminal to the signal source. Most board vendors re-use an existing digital channel for the counter input.

As an example of how Counter may be used, consider a sensor that generates a sinusoid with a frequency that corresponds to the velocity of a fluid. Typical frequencies can range from 1 to 10 kHz. Assuming that you have set the simulation step size to 0.1 sec, the number of pulses that occur in one step of the simulation can be computed by dividing the counter value (output of the rt-DataIn block configured to the counter) by the simulation time step.

### **Current channel class**

The Current parameter supports current output applications. It is set to a range of 4 - 20 mA, which provides a full 12 bits of resolution over the range.

This channel class appears only if your I/O board has current.

### **Digital channel class**

The Digital parameter provides an ON/OFF channel input. When Digital is activated, the input behaves like a current sink. When the voltage level on the digital input line goes low, current flows from the 5 V power supply to ground. When the digital input channel is activated, the voltage level of the channel goes low (turns off). Most digital input channels are capable of sinking 10 mA.

For digital inputs, the rt-DataIn block uses the vendor's specifications for TTL-level values.

### **Thermocouple channel class**

The Thermocouple parameter provides thermocouple linearization input. A thermocouple produces a voltage corresponding to the temperature measured.

When Thermocouple is activated, the Channel Type box shows different types of thermocouples (B, E, J, K, R, S, and T).

VisSim/Real-TimePRO provide for cold junction compensation for the thermocouple linearization; it uses channel 7 on the I/O board for reading the temperature from the solid state temperature device on the multiplexer board for cold junction compensation.

### **Volts channel class**

The Volts parameter provides a time varying voltage input. The range of the voltage input is software-selectable for many boards. Boards with this characteristic are referred to as *programmable gain* boards. If you are using a programmable gain

board and you activate Volts, the voltage ranges are presented under the Channel Type box. These typically range from  $\pm 10$  to  $\pm 0.01$  V and lower.

For boards without programmable gain, the input voltage range is normally set using a micro-switch on the I/O board itself. The corresponding voltage ranges, listed under the Channel Type box, are read-only settings.

The voltage channel is often called an A/D channel. Analog input signals are converted to digital representations using a converter. The converter consists of registers whose numbers determine the resolution. Most boards use 12-bit resolution converters; however, some boards use higher resolution converters for greater precision.

Channel resolution is proportional to channel read time. The higher the channel resolution, the longer it takes to read the channel. In cases where accuracy is less important than speed, you may reduce the resolution.

## Specifying a channel type

The channel type varies according to the possible input ranges for the board. It is also dependent on your selection under Channel Class.

When you select this channel class	Channel Type contains this information
Counter	No channel types are listed. A special internal channel type is automatically selected.
Current	Allowable ranges.
Digital	No channel types are listed. A special internal channel type is automatically selected.
Thermocouple	Allowable thermocouple types: J: Iron/Constantan K: CHROMEGA-ALOMEGA T: Copper/Constantan E: CHROMEGA/Constantan R: Pt 13% Rh/Pt S: Pt 10% Rh/Pt B: Pt 6% Rh/Pt 30% Rh
Volts (for a programmable gain board)	Allowable ranges.
Volts (for a non-programmable gain board)	Hardware settings for the input voltage range on the board.

## Specifying cold junction

Cold junction compensation works in conjunction with a thermocouple. A thermocouple is a temperature measurement sensor that consists of two dissimilar metals joined together at one end to form a junction. When the junction is heated, a small thermoelectric voltage is produced. Cold junction compensation compensates for the voltage generated on the mux board and subtracts it from the voltage at the junction. The rt-DataIn block uses the vendor's specification for the cold junction compensation. Typically, it is 0. If you physically change the cold junction compensation on the mux board, use this parameter to make the same change to the software.

The Cold Junction Compensation parameter is available for use only when the Thermocouple under Channel Class is activated.

---

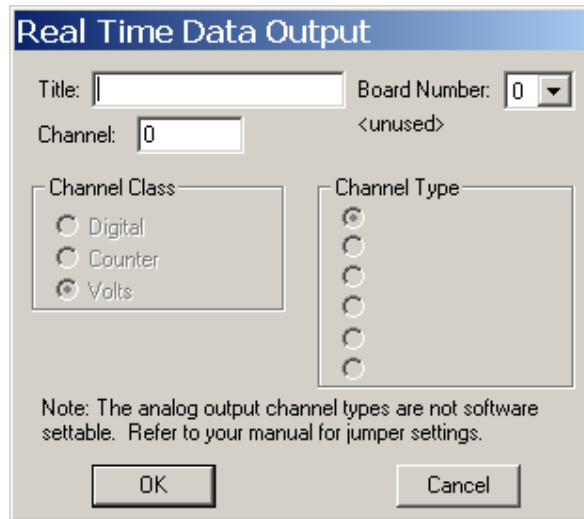
## Using the rt-DataOut block

You use the rt-DataOut block to set an output channel on your I/O board.

### To set up an output channel

1. Choose **Blocks > Real Time** and drag an **rt-DataOut** block into the work area.
2. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the **rt-DataOut** block.
  - Click the right mouse button over the **rt-DataOut** block.

The Real Time Data Output dialog box appears.



3. Make the appropriate selections. (See the descriptions below for more information about each parameter.)
4. Click on the **OK** button, or press **ENTER**.

### Specifying the board number

In the Board Number box, enter the board number of the I/O board whose output channel is to be configured. The board number was established with the [File > Real Time Config command](#). If you are not sure of the board number, click on the DOWN ARROW to reveal the list of configured board numbers. When you click on a board number, the board type is displayed beneath the board number.

### Specifying a channel title

You use the Title box to create titles for each output channel. If you are using more than one output channel on your I/O board, it is a good idea to use channel titles to distinguish each channel. Channel titles appear on the rt-DataOut blocks.

### Specifying a channel number

In the Channel box, you enter a number that correspond to the channel number on the screw terminal or termination panel supplied with your I/O board.

VisSim/Real-TimePRO use channel 0 as the first channel, even if the documentation supplied by the board vendor describes the first channel as channel 1.

[8255-based digital I/O boards](#): If you are using an 8255-based digital I/O board, you can connect `rt-DataOut` channels to electrical connector terminals in several different ways. Boards with an 8255 chip are listed below:

Vendor	Board
Keithley MetraByte	PIO-12
	PIO-24
Measurement Computing	CIO-DIO24
	CIO-DIO24H
National Instruments	PC-DIO-24
	PC-DIO-96
	DAQCard-1200

[PCM-CIA DAS 12/16 and DAS 16/16 boards](#): If you are connecting to a Measurement Computing PCM-CIA DAS 12/16 or DAS 16/16, you can connect the `rt-DataOut` channels to electrical terminals in a specific way.

## Specifying a channel class

The `rt-DataOut` block recognizes three classes of output channels:

- Analog
- Digital
- Pulse

In general, when you select one of these channel classes, it is assumed that you have connected the hardware line to the specified channel number.

When you select a channel class, the options listed under the Channel Type box change accordingly. For example, when you select Volts, the voltage channel types appear under Channel Type.

### **Counter channel class**

The Counter parameter outputs high frequency square waves. The counter output utilizes the board's counter timer and often re-uses an existing digital I/O channel. Refer to the documentation accompanying your board for information on wiring this channel. Data sampling rates of up to 20 KHz (counter-assisted) can be achieved.

As an example of how the counter output operates, consider the control of a stepper motor/drive system. The drive is capable of receiving command pulses from 0 to 5 kHz to regulate its speed. By connecting a slider block, with a range of 0 to 5,000, to the `rt-DataOut` block, configured for counter output, the motor speed can be controlled over its full range.

### **Digital channel class**

The Digital parameter provides an ON/OFF channel output. When Digital is activated, the output behaves like a current sink. When the voltage level on the digital output line goes low, current flows from the 5 V power supply to ground. When the digital output channel is activated, the voltage level of the channel goes

low (turns off). Most digital output channels are capable of sinking 10 mA. For real-time digital output, the inputs generated are Boolean in nature.

The voltages corresponding to the ON/OFF states of the digital channel obey TTL-level values. Low is less than 0.7 V, and high is greater than 2.5 V.

Digital I/O lines are especially useful for controlling equipment using ON/OFF signals. Most often, the channel itself will not have enough power to actuate. In these situations, an opto-isolated solid-state relay is used. These relays, supplied by the board vendor, are soldered onto the screw terminal panel. Instructions are provided by the board vendor.

The opto-isolators have a low power side and a high power side. The digital I/O communicates with the low power side. By connecting a high power source plus equipment through the high power side, you can switch high AC or DC power using the digital I/O line.

Most vendors offer opto-isolator modules ranging from 3 A, with voltages to 280 V AC, to 45 A, with voltages to 650 V AC.

### **Volts channel class**

The Volts parameter provides a time varying voltage output. The range of the voltage output is software-selectable for many boards. Boards with this characteristic are referred to as *programmable gain* boards. If you are using a programmable gain board and you activate Volts, the voltage ranges are presented under the Channel Type box. These typically range from  $\pm 10$  to  $\pm 0.01$  V and lower.

For boards without programmable gain, the output voltage range is normally set using a micro-switch on the I/O board itself. The corresponding voltage ranges, listed under the Channel Type box, are read-only settings.

The voltage channel is often called a D/A channel. Analog input signals are converted to digital representations using a converter, which consists of registers whose numbers determine the resolution. Most boards use 12-bit resolution converters; some boards use higher resolution converters for greater precision.

Channel resolution is proportional to channel read time. The higher the channel resolution, the longer it takes to read the channel. In cases where accuracy is less important than speed, you may reduce the resolution.

## **Specifying a channel type**

The channel type varies according to the possible output ranges for the I/O board.

<b>When you select this channel class</b>	<b>The Channel Type contains this information</b>
Counter (for programmable gain or range board)	There are no channel types listed. A special internal channel type is automatically selected.
Counter (for non-programmable gain or range board)	The range value specified with the File > Real Time Config command.
Digital	A special internal type. If the board does not support programmable ranges or gains, the channel type is set to the range value specified with the File > Real Time Config command. There are no other channel options. There are no other channel options.

Volts (for a programmable gain board)	The allowable ranges from which you can choose. These are read-only settings.
Volts (for non-programmable gain board)	The hardware settings for the output voltage range on the board. These are read-only settings.



# Using the NI-DAQmx Blocks

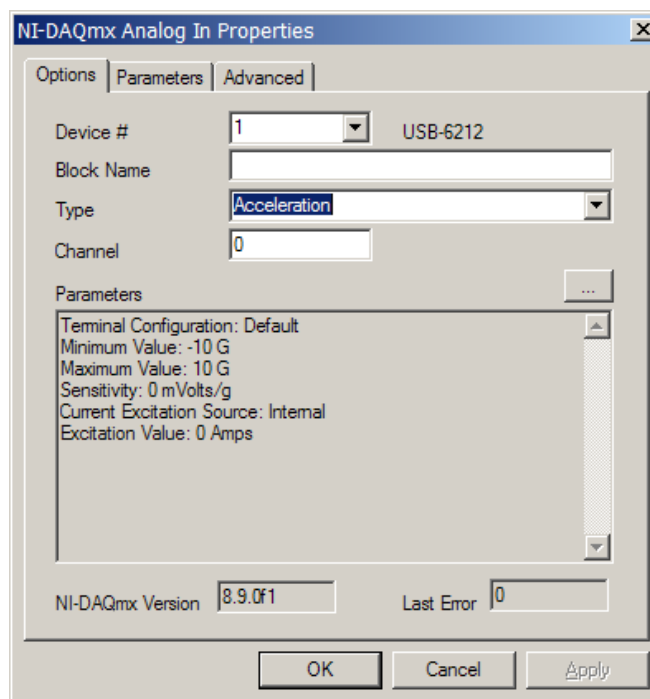
This section describes how to configure the NI-DAQmx blocks. The block options are described in detail in the National Instruments NI-DAQmx C Reference Help file. When you install your National Instruments boards, this help is installed in Program Files\National Instruments\NI-DAQ\Docs\cdaqmx.chm.

---

## NI-DAQmx Analog In

The NI-DAQmx Analog In block lets you configure input channels on your I/O board.

### Options



**Device #:** Indicates the device number.

**Block Name:** Indicates the name that will appear on the block. When nothing is specified, the block name is “Analog-Device-Number/Parameter-Type.”

**Type:** Determines the parameter type to be set or displayed. Click on the down arrow for a list of settable parameter types. To edit a parameter type, select it from the drop-down list, then click on the Parameters tab.

**Parameters:** Read-only text box that lists the current parameter type and corresponding values.

**NI-DAQmx Version:** Indicates the NI-DAQmx version.

**Last Error:** Indicates the last error.

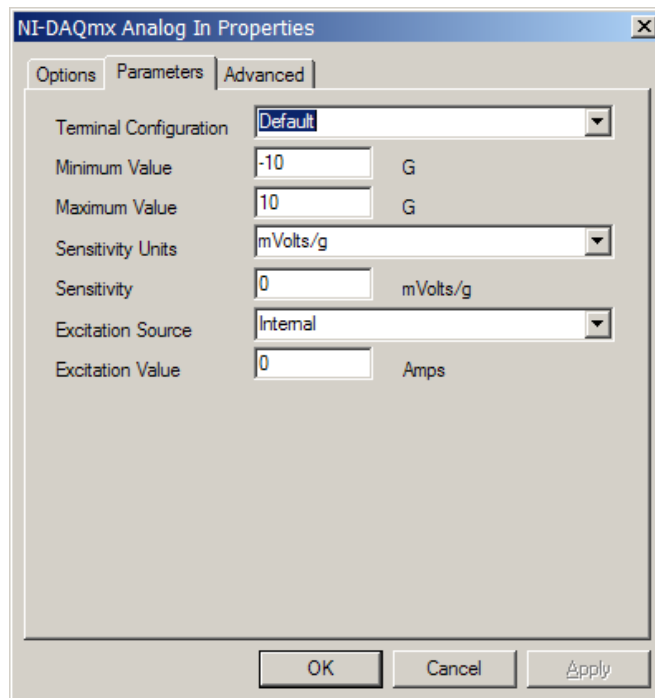
## Parameters

This tab lets you edit parameter values for each parameter type.

### **Acceleration**

Creates channel(s) that use an accelerometer to measure acceleration.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\AccelChan.



**Terminal Configuration:** Indicates the input terminal to be configured for the channel.

**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Sensitivity Units:** Indicates the units of sensitivity.

**Sensitivity:** Indicates the sensitivity of the sensor. The value is in the units you specify in Sensitivity Units.

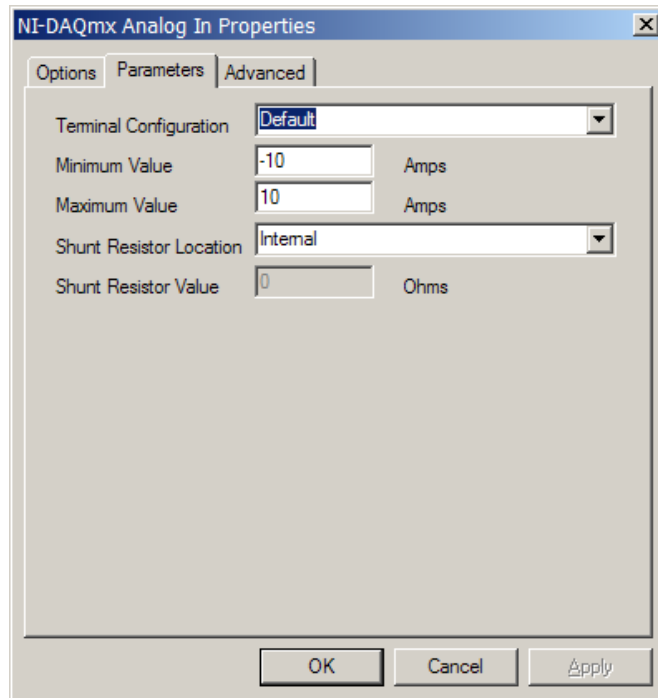
**Excitation Source:** Indicates the source of the excitation.

**Excitation Value:** Indicates the amount of excitation, in amperes, that the sensor requires.

## **Current**

Creates channel(s) for current measurement.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\CurrentChan.



**Terminal Configuration:** Indicates the input terminal to be configured for the channel.

**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

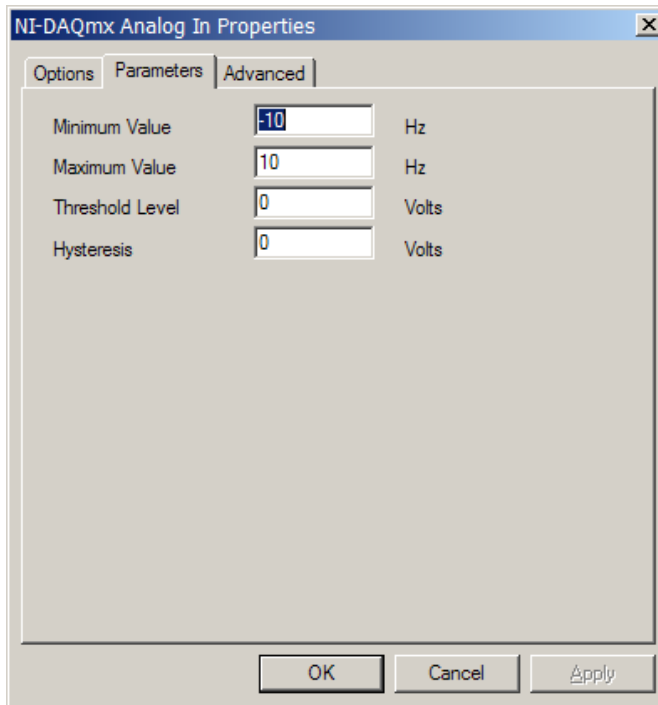
**Shunt Resistor Location:** Indicates the location of the shunt resistor.

**Shunt Resistor Value:** Indicates the value, in ohms, of an external shunt resistor.

## **Frequency**

Creates channel(s) that use a frequency-to-voltage converter.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\FreqVoltageChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

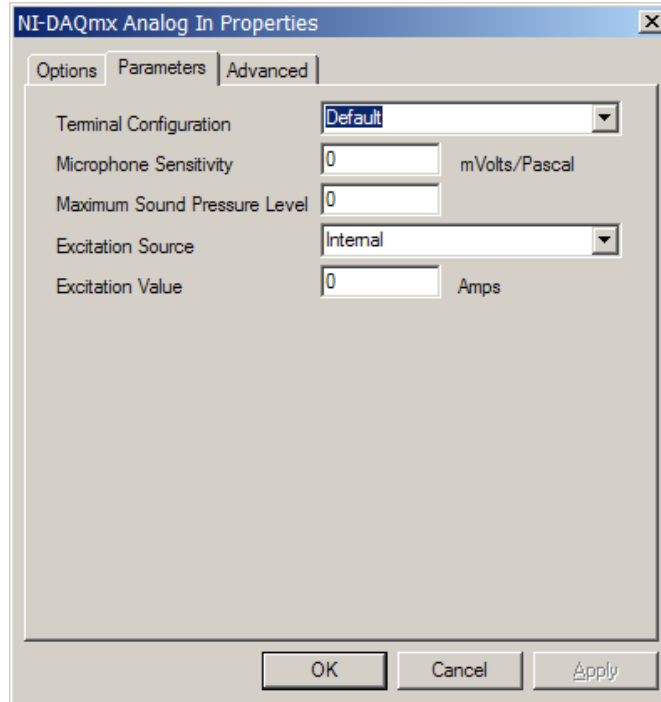
**Threshold Level:** Indicates the voltage level at which to recognize waveform repetitions.

**Hysteresis:** Specifies the volts in a window below the threshold level.

### ***Sound Pressure (Microphone)***

Creates channel(s) that use a microphone to measure sound pressure.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\MicrophoneChan.



**Terminal Configuration:** Indicates the input terminal to be configured for the channel.

**Microphone Sensitivity:** Indicates the sensitivity of the microphone.

**Maximum Sound Pressure Level:** Indicates the maximum instantaneous sound pressure level you expect to measure.

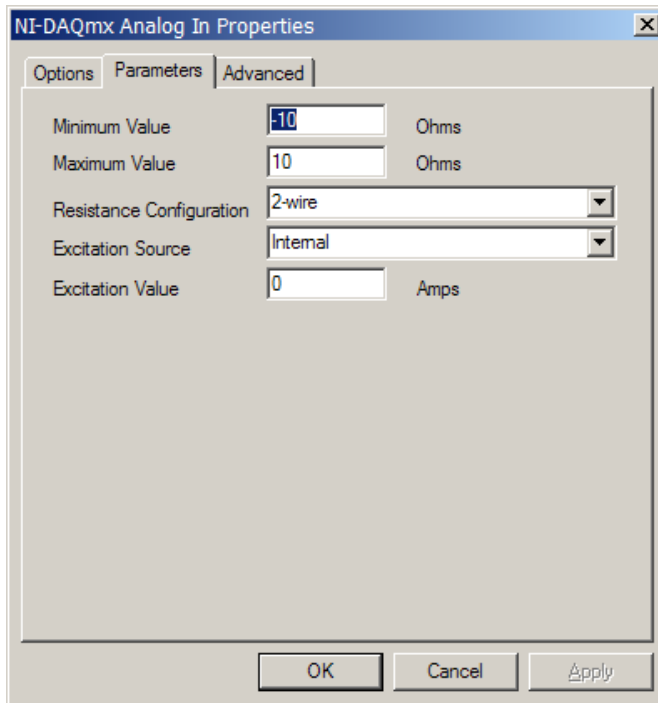
**Excitation Source:** Indicates the source of the excitation.

**Excitation Value:** Indicates the amount of excitation, in amperes, that the sensor requires.

### ***Resistance***

Creates channel(s) to measure resistance.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\ResistanceChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Resistance Configuration:** Indicates the configuration for resistance measurements.

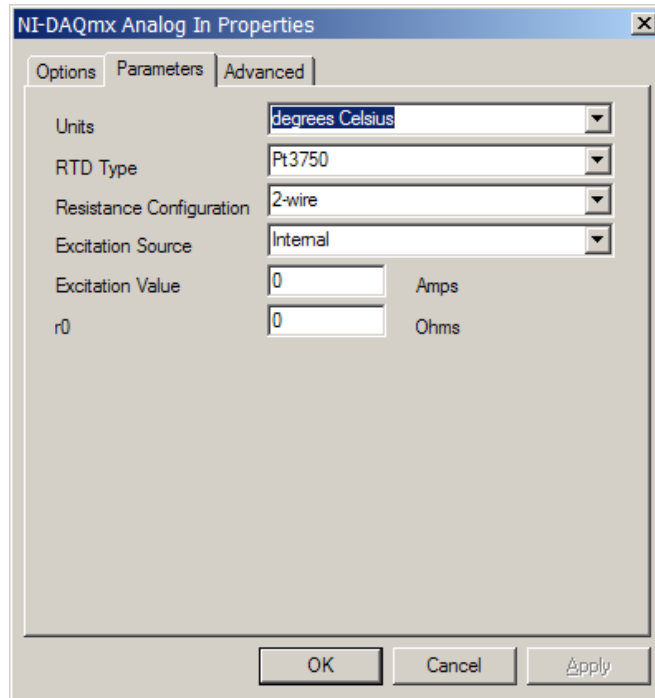
**Excitation Source:** Indicates the source of the excitation.

**Excitation Value:** Indicates the amount of excitation, in amperes, that the sensor requires.

### ***Temperature/RTD***

Creates channel(s) that use RTD to measure temperature.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\RTDChan.



**Units:** Indicates the units to be used to return the measurement.

**RTD Type:** Indicates the type of RTD connected to the channel.

**Resistance Configuration:** Indicates the configuration for resistance measurements.

**Excitation Source:** Indicates the source of the excitation.

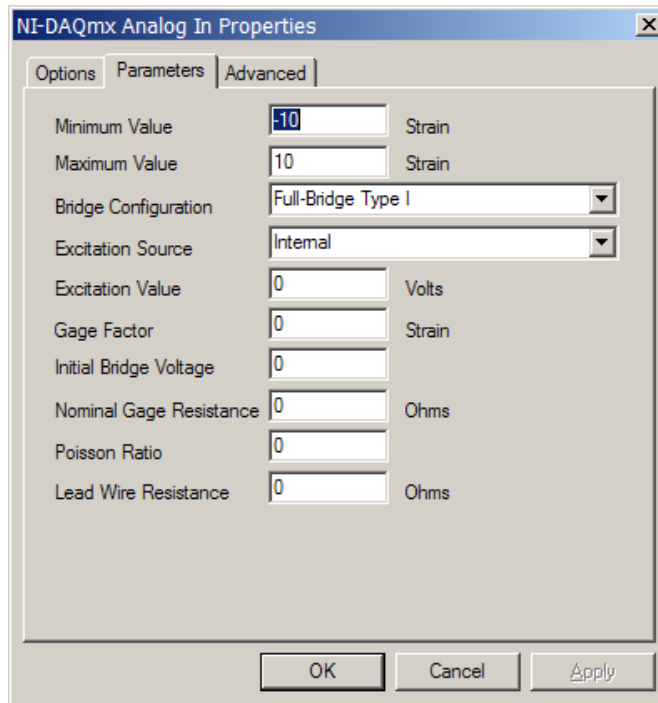
**Excitation Value:** Indicates the amount of excitation, in amperes, that the sensor requires.

**r0:** Indicates the sensor resistance in ohms at 0 deg C for the Callendar-Van Dusen equation.

### **Strain**

Creates channel(s) to measure strain.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\StrainGageChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Bridge Configuration:** Indicates the configuration for strain gage bridge.

**Excitation Source:** Indicates the source of the excitation.

**Excitation Value:** Indicates the amount of excitation, in amperes, that the sensor requires.

**Gage Factor:** Indicates the sensitivity of the strain gages and relates the change in electrical resistance to the change in strain.

**Initial Bridge Voltage:** Indicates the bridge output voltage in the unloaded condition.

**Nominal Gage Resistance:** Indicates the resistance, in ohms, of the gages in an unstrained position.

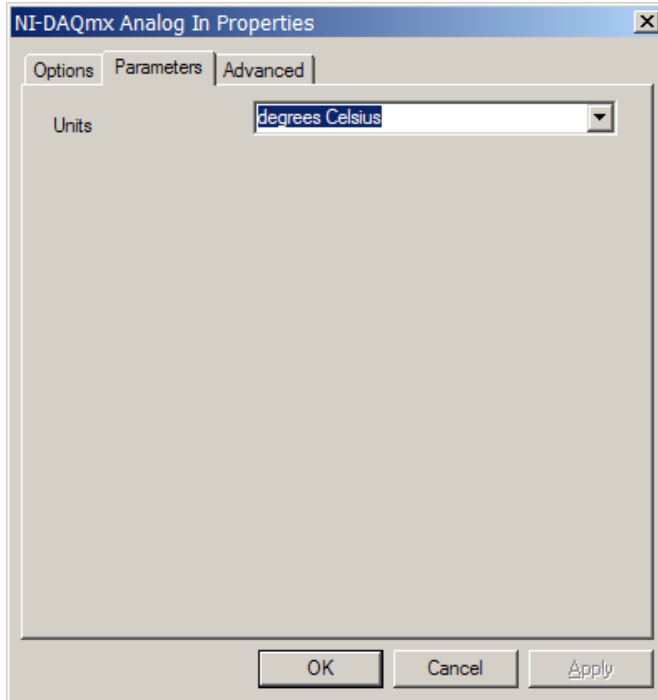
**Poisson Ratio:** Indicates the ratio of lateral strain to axial strain in the material in which you measure strain.

**Lead Wire Resistance:** Indicates the amount, in ohms, of resistance in the lead wires. Ideally, this value is the same for all leads.

### ***Temperature/Built-In Sensor***

Creates channel(s) to measure temperature with a built-in sensor of a terminal block or device.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\TempBuiltInSensorChan.

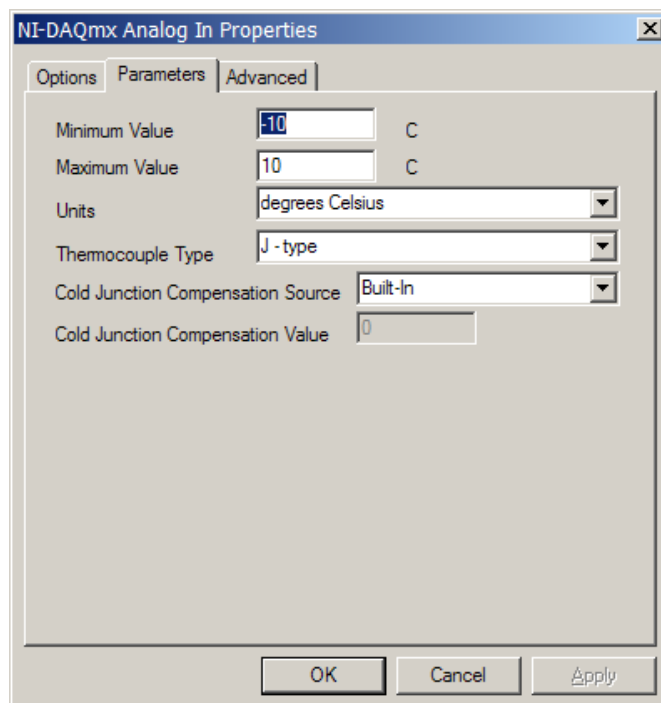


**Units:** Indicates the units to use to return the measurement.

### ***Temperature/Thermocouple***

Creates channel(s) that use a thermocouple to measure temperature.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\ThrmcpplChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Units:** Indicates the units to use to return the measurement.

**Thermocouple Type:** Indicates the type of thermocouple connected to the channel.

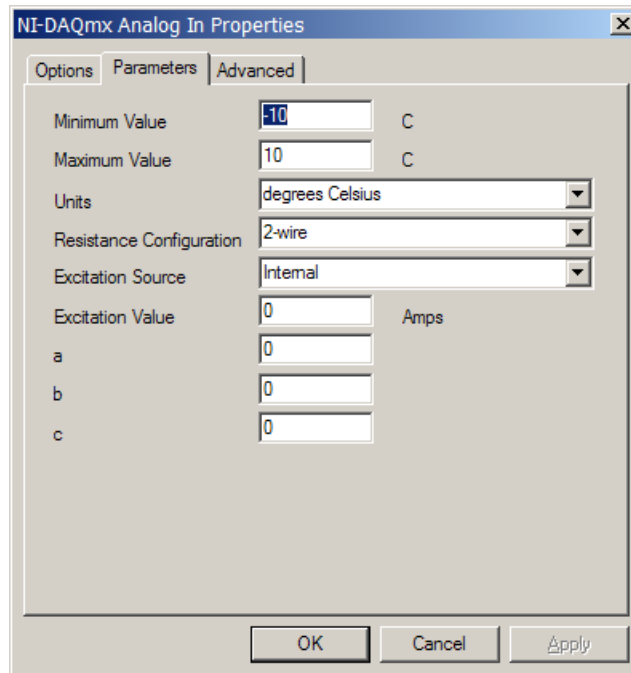
**Cold Junction Compensation Source:** Indicates the source of cold junction compensation.

**Cold Junction Compensation Value:** Indicates the temperature of the cold junction of the thermocouple to which you set the CJC source.

### ***Temperature/Thermistor with Current Excitation***

Creates channel(s) that use a thermistor to measure temperature when the thermistor requires current excitation.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\ThrmstrChanlex.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Units:** Indicates the units to use to return the measurement.

**Resistance Configuration:** Indicates the configuration for resistance measurements.

**Excitation Source:** Indicates the source of the excitation.

**Excitation Value:** Indicates the amount of excitation, in amperes, that the sensor requires.

**a:** Indicates the A constant from the Steinhart-Hart-thermistor equation.

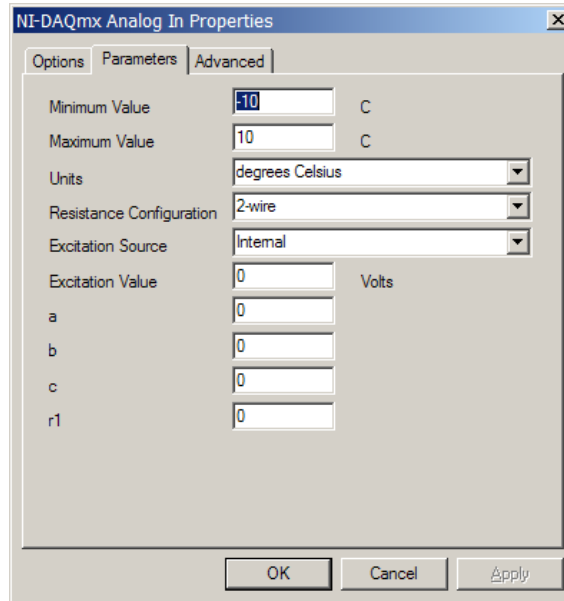
**b:** Indicates the B constant from the Steinhart-Hart-thermistor equation.

**c:** Indicates the C constant from the Steinhart-Hart-thermistor equation.

### **Temperature/Thermistor with Voltage Excitation**

Creates channel(s) that use a thermistor to measure temperature when the thermistor requires voltage excitation.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\ThrmstrChanlvx.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Units:** Indicates the units to use to return the measurement.

**Resistance Configuration:** Indicates the configuration for resistance measurements.

**Excitation Source:** Indicates the source of the excitation.

**Excitation Value:** Indicates the amount of excitation, in amperes, that the sensor requires.

**a:** Indicates the A constant from the Steinhart-Hart-thermistor equation.

**b:** Indicates the B constant from the Steinhart-Hart-thermistor equation.

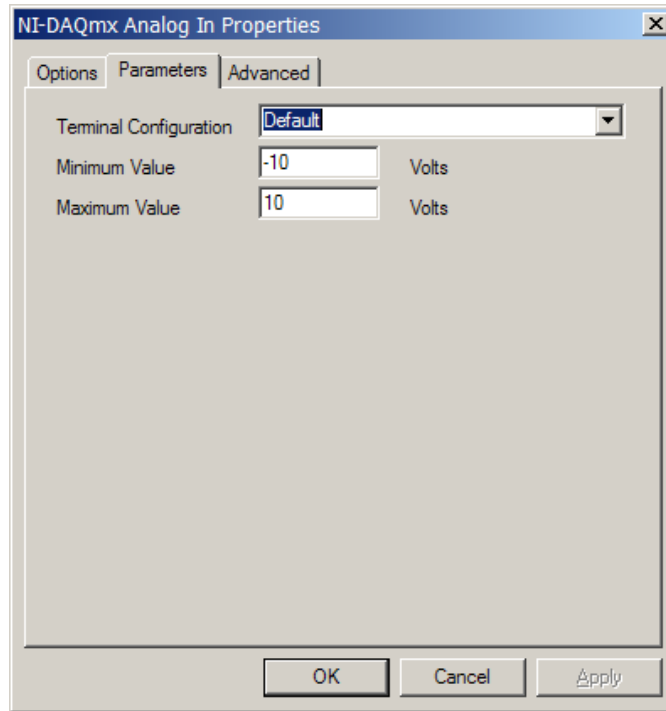
**c:** Indicates the C constant from the Steinhart-Hart-thermistor equation.

**r1:** Indicates the value, in ohms, of the reference resistor.

### **Voltage**

Creates channel(s) to measure voltage.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\VoltageChan.



**Terminal Configuration:** Indicates the input terminal to be configured for the channel.

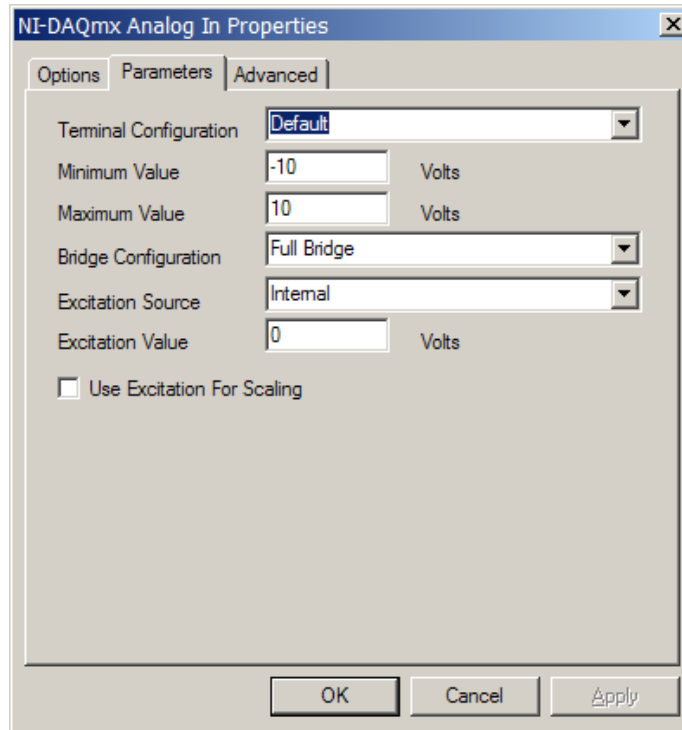
**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

### ***Voltage with Excitation***

Creates channel(s) to measure voltage with excitation to scale the measurement.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Input Channels > DAQmxCreateA\VoltageChanWithExcit.



**Terminal Configuration:** Indicates the input terminal to be configured for the channel.

**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Bridge Configuration:** Indicates the type of Wheatstone bridge.

**Excitation Source:** Indicates the source of the excitation.

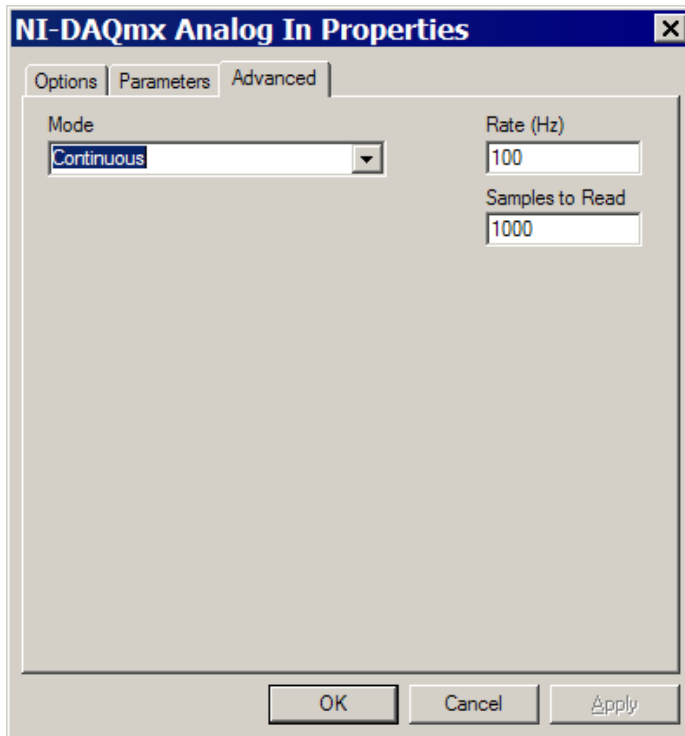
**Excitation Value:** Indicates the amount of excitation, in volts, that the sensor requires.

**Use Excitation for Scaling:** Specifies whether the measurement is divided by the excitation.

## Advanced

This tab lets you edit advanced clocking. Preparing a single USB packet takes a significant amount of time; however, reading or writing the packet is relatively fast. If you acquire data by a single value (that is, specify On Demand), then each value is sent as a single USB packet and the sample rate is significantly slower.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options.



### Mode

**Continuous:** You provide your own clock rate for data acquisition and number of samples to read. In this case you have two output pins: the actual size and the vector of values. You need to examine the size, if it is 0, the data is not there. When the data arrived, the size will be the size of the vector. Every time, VisSim acquires the specified number of samples, the data vector is assigned to the second output pin.

**Finite:** VisSim performs a single read task. When reading is finished, the vector of data is assigned to the second pin and you can examine it in VisSim.

**On Demand:** VisSim reads one value on each simulation step. The clock rate is that of VisSim (that is, under Simulation > Simulation Properties > Range, you activate Run in Real Time). The On Demand mode serves well for the most tasks; however, if the sampling rate is not enough, you may need to use the other modes, which only work with devices that support buffered tasks

**Rate:** Indicates the clock rate.

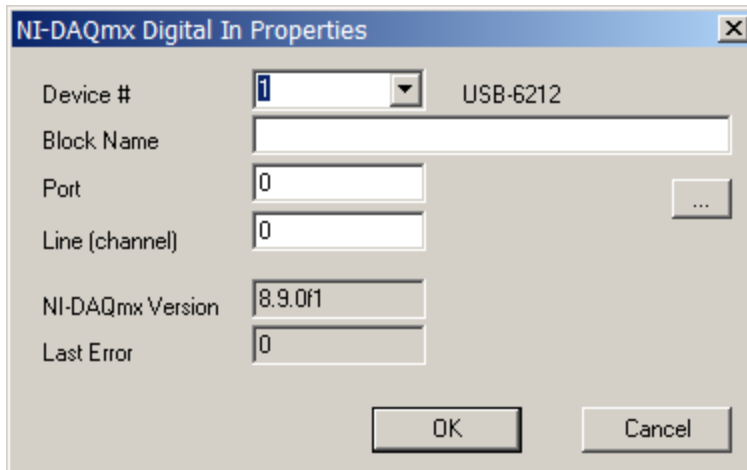
**Samples to Read:** Indicates the number of samples to read.

---

## NI-DAQmx Digital In

The NI-DAQmx Digital In block lets you set input channels to measure digital signals.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Digital Input Channels > DAQmxCreateDIChan.



**Device #:** Indicates the device number.

**Block Name:** Indicates the name that will appear on the block. When nothing is specified, the block name is “Digital-Device-Number.”

**Port:** Indicates the port, starting at 0.

**Line:** Indicates a number or expression showing a range of bits in the port.

**NI-DAQmx Version:** Indicates the NI-DAQmx version.

**Last Error:** Indicates the last error.

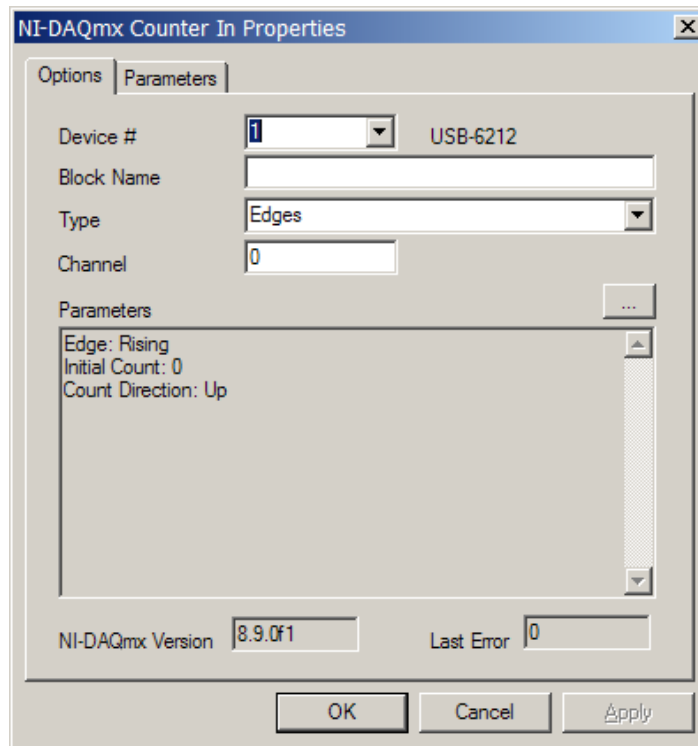
---

## NI-DAQmx Counter In

The NI-DAQmx Counter In block lets you create channels to count the number of rising or falling edges of a digital signal, measure the frequency of a digital signal, measure the period of a digital signal, measure the width of a digital pulse, measure the time between state transitions of a digital signal, and measures the amount of time between the rising or falling edge of one digital signal and the rising or falling edge of another digital signal.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Input Channels.

## Options



**Device #:** Indicates the device number.

**Block Name:** Indicates the name that will appear on the block. When nothing is specified, the block name is “Counter/Device-Number/Parameter-Type.”

**Type:** Determines the parameter type to be set or displayed. Click on the down arrow for a list of settable parameter types. To edit a parameter type, select it from the drop-down list, then click on the Parameters tab.

**Channel:** Identifies the counter that is to be used.

**Parameters:** Read-only text box that lists the current parameter type and corresponding values.

**NI-DAQmx Version:** Indicates the NI-DAQmx version.

**Last Error:** Indicates the last error.

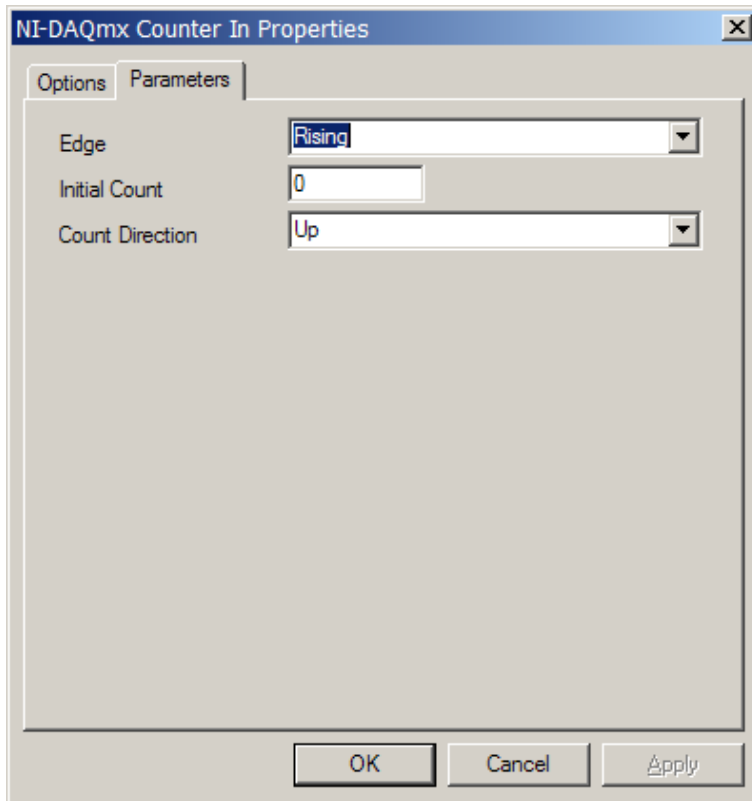
## Parameters

This tab lets you edit parameter values for each parameter type.

### *Edges*

Creates a channel to count the number of rising or falling edges of a digital signal.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Input Channels > DAQmxCreateCICountEdgesChan.



**Edge:** Specifies on which edges of the input signal to increment or decrement the count.

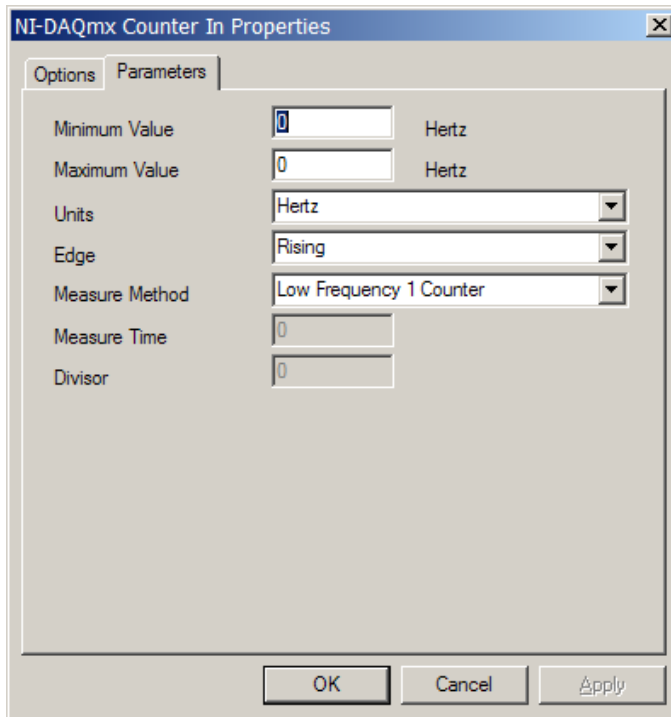
**Initial Count:** Indicates the value from which to start counting.

**Count Direction:** Specifies whether to increment or decrement the counter on each edge.

### ***Frequency***

Creates a channel to measure frequency of a digital signal.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Input Channels > DAQmxCreateCIFreqChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Units:** Indicates units to use to return the measurement.

**Edge:** Specifies between which edges to measure the frequency or period of the signal.

**Measure Method:** Indicates the method used to calculate the period or frequency of the signal.

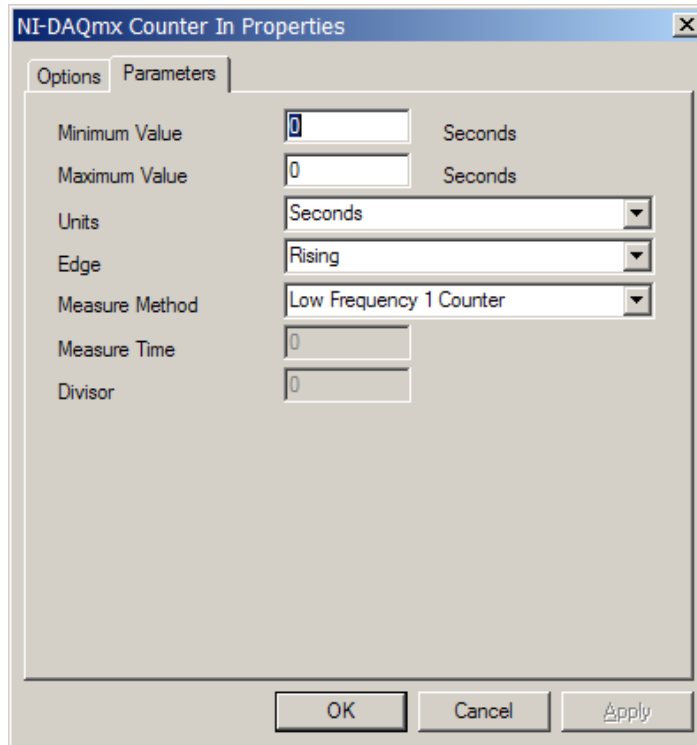
**Measure Time:** Indicates the length of time to measure the frequency or period of a digital signal.

**Divisor:** Indicates the value by which to divide the input signal.

### ***Period***

Creates a channel to measure the period of a digital signal.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Input Channels > DAQmxCreateCIPeriodChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Units:** Indicates units to use to return the measurement.

**Edge:** Specifies between which edges to measure the frequency or period of the signal.

**Measure Method:** Indicates the method used to calculate the period or frequency of the signal.

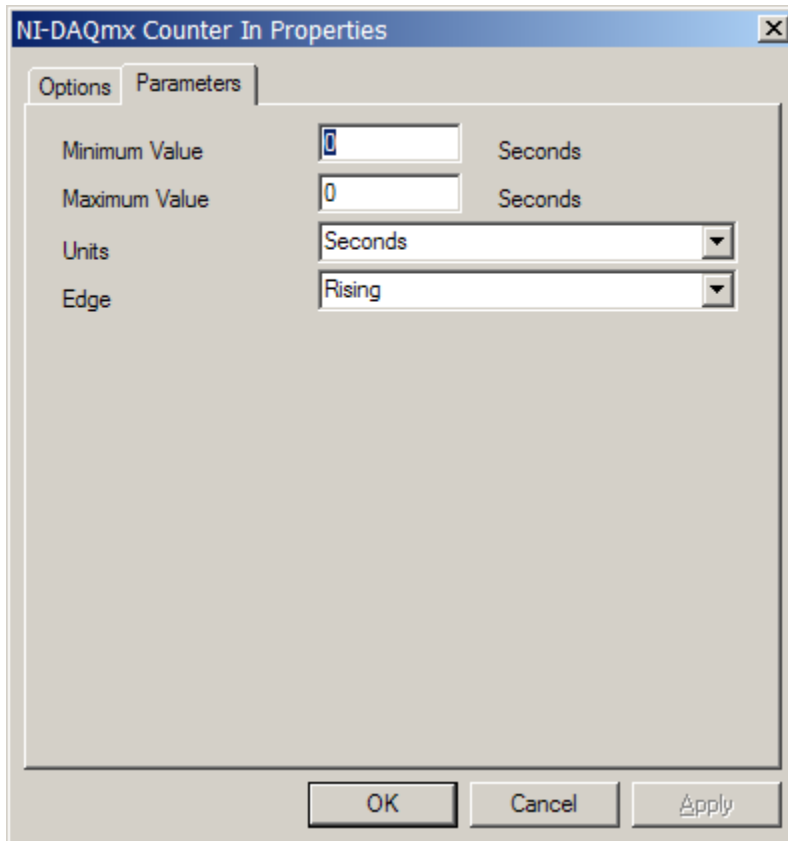
**Measure Time:** Indicates the length of time to measure the frequency or period of a digital signal.

**Divisor:** Indicates the value by which to divide the input signal.

### ***Pulse Width***

Creates a channel to measure the width of a digital pulse.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Input Channels > DAQmxCreateCIPulseWidthChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

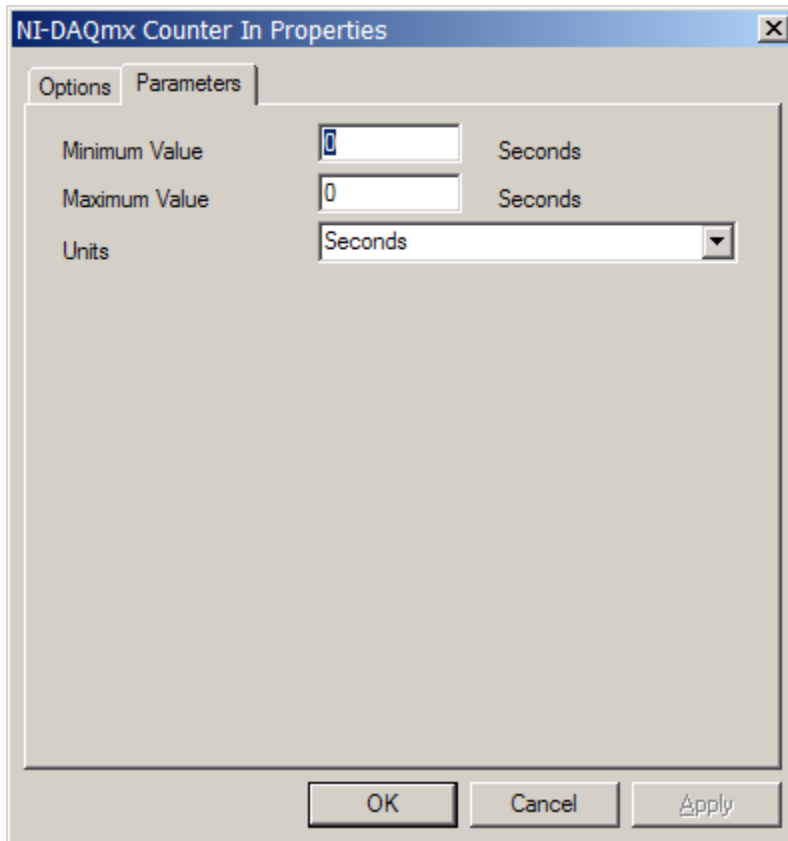
**Units:** Indicates units to use to return the measurement.

**Edge:** Specifies between which edges to measure the frequency or period of the signal.

### ***Semi Period***

Creates a channel to measure the time between state transitions of a digital signal.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Input Channels > DAQmxCreateCISemiPeriodChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

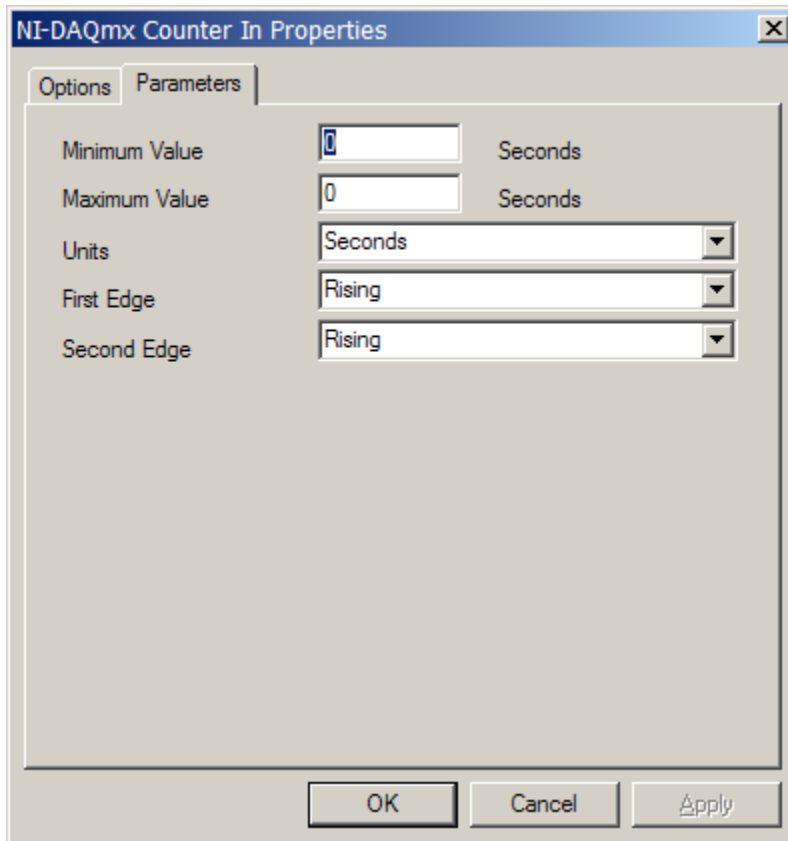
**Maximum Value:** Indicates the maximum value you expect to measure.

**Units:** Indicates units to use to return the measurement.

### ***Time Between Edges***

Creates a channel to measure the amount of time between rising or falling edges of one digital signal.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Input Channels > DAQmxCreateCITwoEdgesSepChan.



**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**Units:** Indicates units to use to return the measurement.

**First Edge:** Specifies on which edge of the first signal to start each measurement.

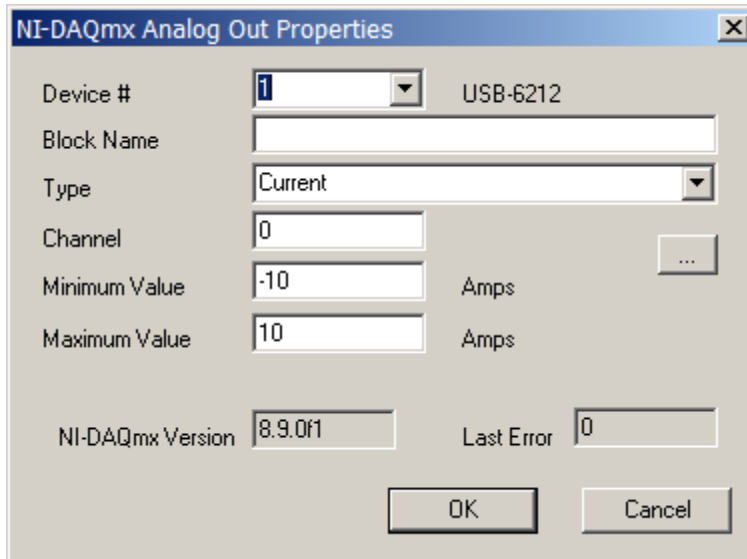
**Second Edge:** Specifies on which edge of the first signal to stop each measurement.

---

## NI-DAQmx Analog Out

The NI-DAQmx Analog Out block lets you generate current or voltage.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Output Channels.



**Device #:** Indicates the device number.

**Block Name:** Indicates the name that will appear on the block. When nothing is specified, the block name is “Digital-Device-Number.”

**Type:** Indicates whether you want to generate current or voltage.

**Channel:** Indicates the name of the physical channel.

**Minimum Value:** Indicates the minimum value you expect to measure.

**Maximum Value:** Indicates the maximum value you expect to measure.

**NI-DAQmx Version:** Indicates the NI-DAQmx version.

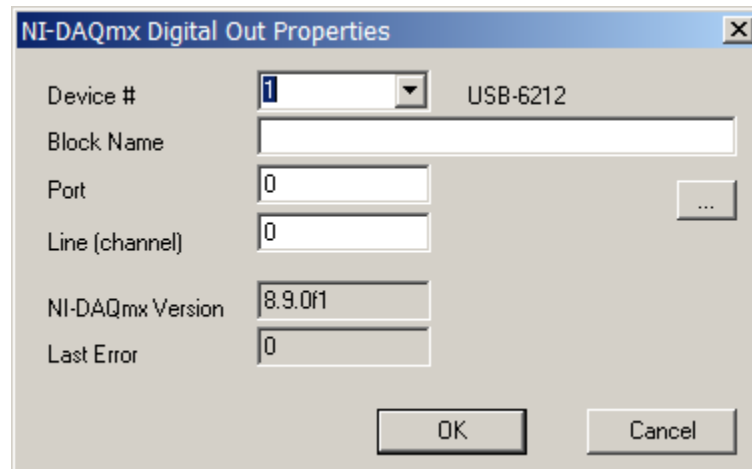
**Last Error:** Indicates the last error.

---

## NI-DAQmx Digital Out

The NI-DAQmx Digital Out block lets you digital signals.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Analog Output Channels.



**Device #:** Indicates the device number.

**Block Name:** Indicates the name that will appear on the block. When nothing is specified, the block name is “Digital-Device-Number.”

**Port:** Indicates the port, starting at 0.

**Line:** Indicates a number or expression showing a range of bits in the port.

**NI-DAQmx Version:** Indicates the NI-DAQmx version.

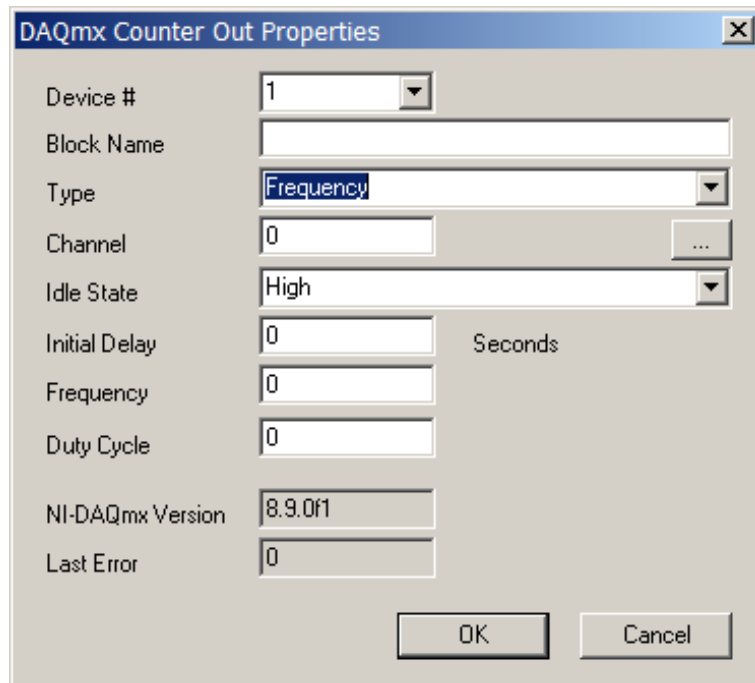
**Last Error:** Indicates the last error.

---

## NI-DAQmx Counter Out

The NI-DAQmx Counter Out block lets you set channels to generate digital pulses.

Please refer to the National Instruments NI-DAQmx C Reference Help file for detailed descriptions of these options, under NI-DAQmx C Functions > Channel Configuration/Creation > Create Counter Output Channels.



**Device #:** Indicates the device number.

**Block Name:** Indicates the name that will appear on the block. When nothing is specified, the block name is “Digital-Device-Number.”

**Type:** Indicates whether frequency or time is used to define the high state of the pulse.

**Channel:** Identifies the counter that is to be used.

**Idle State:** Indicates the resting state of the output terminal.

**Initial Delay:** Indicates the amount of time in seconds to wait before generating the first pulse.

**Frequency:** Indicates the frequency at which to generate pulses.

**Duty Cycle:** Indicates the width of the pulse divided by the pulse period. NI-DAQmx uses this ratio, combined with frequency, to determine pulse width and the interval between pulses.

**NI-DAQmx Version:** Indicates the NI-DAQmx version.

**Last Error:** Indicates the last error.

---

## Reset command

The Reset command returns the device to an initialized state.



# Using the Measurement Computing Universal Driver Blocks

This section describes how to use the MCC Input and MCC Output blocks to communicate with the Measurement Computing universal driver. For the complete list of Measurement Computing boards, contact Measurement Computing.

---

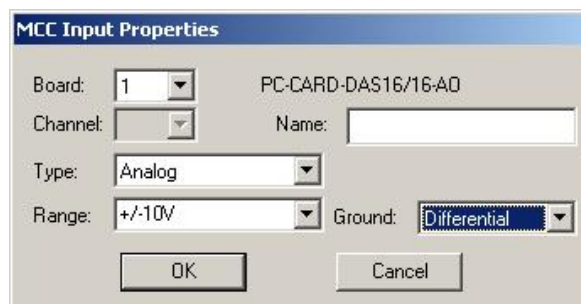
## Using the MCC Input block

Through the Setup dialog box for the MCC Input block, you define the base address and channel number to which the pulse width modulated waveform is sent. You also use the Setup dialog box to set the clock frequency and to provide the PMW block with a unique name.

### To specify MCC Input block properties

1. Choose **Blocks > Real Time** and drag a **MCC Input** block into the work area.
2. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the **MCC Input** block.
  - Click the right mouse button over the **MCC Input** block.

The MCC Input Properties dialog box appears.



3. Do one or more of the following:

Use this parameter	To
Board #	Select the installed Measurement Computing board. Text that describes the board will appear to the right.
Channel	Indicate the analog, digital, or counter channel. For each 8255 unit, channels 0 – 7 correspond to Port A; channels 8 – 15 correspond to Port B; channels 16 – 23 correspond to Port C. Note that multiple digital devices are treated like one large device. Thus digital channel 24 corresponds to Port A bit 0 on the second 8255 digital device.
Type	Indicates whether the board is analog, digital, or counter channels. The channel count is adjusted to reflect the channel type.
Name	Provide a unique name for your MCC <b>Input</b> block.
Range	Indicate the voltage range for the I/O board.
Ground	Indicate whether the I/O board is single-ended (that is, single ground for all channels) or differential (that is, floating ground). If you choose differential, the channel count is adjusted to reflect the channel type.

4. Click on the **OK** button, or press **ENTER**.

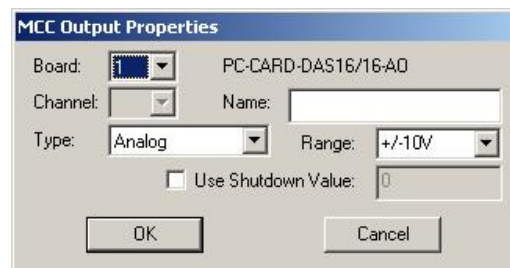
## Using the MCC Output block

Through the Setup dialog box for the MCC Output block, you define the base address and channel number to which the pulse width modulated waveform is sent. You also use the Setup dialog box to set the clock frequency and to provide the PWM block with a unique name.

### To specify MCC Output block properties

1. Choose **Blocks > Real Time** and drag a **MCC Output** block into the work area.
2. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the **MCC Output** block.
  - Click the right mouse button over **the MCC Output** block.

The 9513 PWM Counter Output Setup dialog box appears.



3. Do one or more of the following:

Use this parameter	To
Board #	Select the installed Measurement Computing board. Text that describes the board will appear to the right.

Channel	Indicate the analog, digital, or counter channel. For each 8255 unit, channels 0 – 7 correspond to Port A; channels 8 – 15 correspond to Port B; channels 16 – 23 correspond to Port C. Note that multiple digital devices are treated like one large device. Thus digital channel 24 corresponds to Port A bit 0 on the second 8255 digital device.
Type	Indicates whether the board is analog, digital, or counter channels. The channel count is adjusted to reflect the channel type.
Name	Provide a unique name for your MCC <b>Input</b> block.
Range	Indicate the voltage range for the I/O board.
Use Shutdown Value	Present the indicated value to the output port when the simulation is stopped.

4. Click on the **OK** button, or press **ENTER**.



# Producing Pulse Width Modulated Output

This section contains...

---

## Boards supported by the PWM block

The PWM block is based on the 9513 chip. Any board based on that chip can use the PWM block. Currently, these boards include:

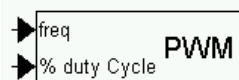
Vendor	Board
Analog Devices	RTI 815
Measurement Computing	CIO-CTR05
	CIO-CTR05/H50
	CIO-CTR10
	CIO-CTR10/H50

Because VisSim/Real-TimePRO is continually expanding the I/O boards it supports, it's a good idea to contact Visual Solutions for the most up-to-date list of supported boards.

---

## Using the PWM block

The PWM block has two inputs: frequency and % Duty Cycle (or pulse width).



Combined, these two inputs form a signal with the specified frequency and duty cycle. The duty cycle indicates the percent of time the signal is in an ON state. The PWM block sends the signal to the specified channel on the 9513 counter output.

Although you can dynamically change the input values simultaneously, you typically vary one or the other input at a time. For example, for power modulation, keep the frequency fixed at a constant value while varying the pulse width. Conversely, for pure frequency output, fix the pulse width and vary the frequency.

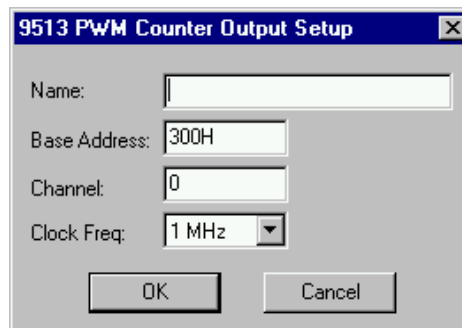
# Specifying properties of the PWM block

Through the Setup dialog box for the PWM block, you define the base address and channel number to which the pulse width modulated waveform is sent. You also use the Setup dialog box to set the clock frequency and to provide the PWM block with a unique name.

## To specify PWM block properties

1. Choose **Blocks > Real Time** and drag a **PWM** block into the work area.
2. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the **PWM** block.
  - Click the right mouse button over the **PWM** block.

The 9513 PWM Counter Output Setup dialog box appears.



3. Do one or more of the following:

Use this parameter	To
Name	Provide a unique name for your PWM block. By naming PWM blocks, you reduce the risk of misreading your diagram, particularly when you are using more than one PWM block.
Base Address	Indicate the I/O port register address through which the real-time driver commands the board. Enter the base address as a hexadecimal number, followed by an optional "H." If you are not sure of the board's base address, use the <a href="#">File &gt; Real Time Config command</a> .
Channel	Enter a number that corresponds with the channel number on the screw terminal or termination panel supplied with your I/O board. VisSim/Real-TimePRO uses channel 0 as the first channel, even if the documentation supplied by the board vendor describes the first channel as channel 1.
Clock Freq.	Enter a value that corresponds with the base frequency crystal value set on your I/O board. If the value you enter does not match the base frequency crystal value, erroneous results are produced.

4. Click on the **OK** button, or press **ENTER**.

# Measuring Frequency Input

This section contains...

---

## Boards supported by the frequency block

The frequency block is based on the 9513 chip. Any board based on that chip can use the frequency block. Currently, these boards include:

Vendor	Board
Analog Devices	RTI 815
Measurement Computing	CIO-CTR05
	CIO-CTR05/H50
	CIO-CTR10
	CIO-CTR10/H50

Because VisSim/Real-TimePRO is continually expanding the I/O boards it supports, it's a good idea to contact Visual Solutions for the most up-to-date list of supported boards.

---

## Using the frequency block

The frequency block lets you access and read a particular channel on the 9513 counter and use the value of this signal in a VisSim diagram. Functionally, the frequency block performs the reverse options of the PWM [block](#).

---

## Specifying properties of a frequency block

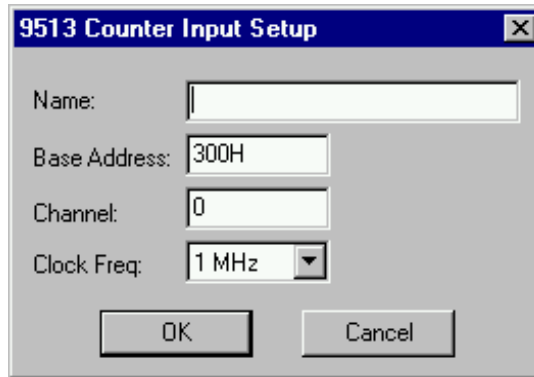
The frequency block measures a signal on an I/O board at the specified base address and channel number and outputs the frequency in hertz.

### To specify frequency block properties

1. Choose **Blocks > Real Time** and drag a **frequency** block into the work area.
2. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the **frequency** block.

- Click the right mouse button over the **frequency** block.

The 9513 Counter Input Setup dialog box appears.



3. Do one or more of the following:

Use this parameter	To
Name	Provide a unique name for your <b>frequency</b> block. By naming frequency blocks, you reduce the risk of misreading your diagram, particularly when you are using more than one frequency block.
Base Address	Indicate the I/O port register address through which the real-time driver commands the board. Enter the base address as a hexadecimal number, followed by an "H." If you are not sure of the board's base address, use the <a href="#">File &gt; Real Time Config command</a> .
Channel	Enter a number that corresponds with the channel number on the screw terminal or termination panel supplied with your I/O board. VisSim/Real-TimePRO uses channel 0 as the first channel, even if the documentation supplied by the board vendor describes the first channel as channel 1.
Clock Freq.	Enter a value that corresponds with the base frequency crystal value set on your I/O board. If the value you enter does not match the base frequency crystal value, erroneous results are produced.

4. Click on the **OK** button, or press **ENTER**.

---

## Specifying a sampling rate

To obtain an accurate measurement, set VisSim's sampling interval (that is, its simulation time step) larger than the sample interval of the events you are measuring.

# Reading Encoder Input

This section contains...

---

## Boards supported by the encoder block

Currently, only the MFIO-3A board from Precision Micro Dynamics is supported. However, because VisSim/Real-TimePRO is continually expanding the I/O boards they support, it is a good idea to contact Visual Solutions for the most up-to-date list of supported boards.

---

## Using the encoder block

Because the MFIO-3A board has quadrature encoder inputs, to access all three encoder inputs, use three encoder blocks.

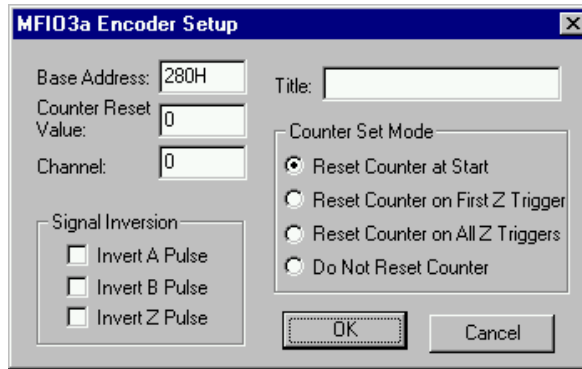
---

## Specifying properties of an encoder block

### To set up the encoder block

1. Choose **Blocks > Real Time** and drag an encoder block into the work area.
2. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the encoder block.
  - Click the right mouse button over the **encoder** block.

The MFIO3a Encoder Setup dialog box appears.



3. Make the appropriate selections. (See the descriptions below for more information about each parameter.)
4. Click on the **OK** button, or press **ENTER**.

## Specifying a title

You use the Title box to create a name for the particular encoder channel that is accessed with the encoder block. By giving unique names to different encoder inputs, you can easily distinguish between them.

## Specifying the base address

The base address indicates the I/O port register address through which the real-time driver commands the board. Enter the base address as a hexadecimal number, followed by an “H.”

If you are not sure of the board’s base address, use the [File > Real Time Config command](#).

## Specifying the counter reset value

The Counter Reset Value box lets you reset the value of the index pulse on the occurrence of the index pulse and according to the counter set mode.

## Specifying the channel

In the Channel box, you enter a number that corresponds with the channel number on the screw terminal or termination panel supplied with your MFIO-3A board. VisSim/Real-TimePRO uses channel 0 as the first channel, even if the documentation supplied by the board vendor describes the first channel as channel 1.

## Specifying the signal inversion

Each channel on the MFIO-3A board has an A, B, and Z axis. You can invert one or more axes to get phasing information.

## Specifying the counter set mode value

The Counter Set Mode lets you reset the counter on your MFIO-3A board to the value specified in the Counter Reset Value box. You have the following choices:

- You can reset the counter at the start of the simulation.

- You can reset the counter at the first occurrences of the index pulse (which is also known as the Z trigger).
- You can reset the counter at every occurrence of the index pulse.

You can alternatively not ever reset the counter.

---

## Determining the direction of rotation value

The two pulses, A and B, are shifted by a quarter of a cycle with respect to each other. The shift between the two signals enables the controller or the simulation to determine the direction of rotation, according to whether pulse A leads pulse B, or vice versa.



# Developing Servo Control Systems

This section describes how to use the Tech80 5650A block to interactively develop servo control systems. Using the Tech80 5650A block, you can control and tune up to four independent axes of closed-loop servo control. There are three methods of closed-loop control: two position-based and one velocity-based. There is also a controller-disabled mode that lets you write to the DAC16 or PWM directly.

---

## Boards supported by the Tech80 5650A block

The Tech80 5650A block supports the Technology 80 5650A servo motor controller board by providing a Technology 80 5650A servo motor controller driver with the software. The 5650A servo motor controller board supports four axes and uses an on-board PID or PIVff controller, together with encoder feedback, to control the motion on each axis. Each axis has a choice of PWM output or 16-bit D-to-A converter that goes from +/-10V.

---

## Using a Tech80 5650A block

Each Tech80 5650A block you use controls a single axis.

### To insert a Tech80 5650A block in your diagram

1. Choose **Blocks > RealTime**.
2. Point to the **Tech80 5650A** block and click the mouse.
3. Point to the location in the diagram where you want to insert the **Tech80 5650A** block and click the mouse.

---

## Tech80 5650A block input

The Tech80 5650A block has one input. The expected input varies in both units and range depending on the profile mode used for the axis. There are four profile modes — Velocity, Trapezoid, S-Curve, and Voltage — and you specify them via the Tech80 Setup dialog box. For more information, see the descriptions of the profile modes under “Configuring a Tech80 Controller.”



---

## Tech80 5650A block outputs

There can be from one to three outputs on the Tech80 5650A block. The top most output is the current actual position of the axis. This position is determined from the encoder feedback and is in units of revolutions. To convert this value to counts, multiply by the number of counts per revolution of the motor.

The second output is the desired position in revolutions. The desired position is the position where the on-board controller is trying to place the axis. Since this position is determined by the on-board controller, it is meaningful only when the on-board controller is used.

The third output is the desired velocity in units of revolutions per second. This is also determined by the on-board controller and is meaningful only when the on-board controller is used.

You can change the number of output connectors using the Edit menu's Add Connector and Remove Connector commands or the  and  toolbar buttons.

---

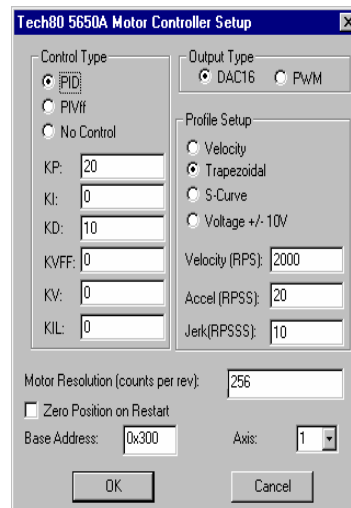
## Configuring a Tech80 5650A block

You configure the Technology 80 5650A controller via the Setup dialog box for the Tech80 5650A block.

### To access the Setup dialog box

1. Do one of the following:
  - Choose **Edit > Block Properties** and click the mouse over the **Tech80 5650A** block.
  - Click the right mouse button over the **Tech80 5650A** block.

The Tech80 5650A Motor Controller Setup dialog box appears.



2. Make the appropriate selections. (See the descriptions below for more information about each parameter.)
3. Click on the **OK** button, or press **ENTER**.

## Specifying a control type

The control type determines the on-board controller to be used by the Technology 80 5650A servo motor controller board. There are three control algorithms and six control modes related to the algorithms.

Control algorithm	Description	Control modes for this algorithm
PID	Specifies a Proportional, Integral, Derivative control algorithm. You can specify PID for all four axes.	The control parameters used with the PID are KP, KI, KD and KIL.
PIVff	Specifies a Proportional, Integral Velocity feed forward control algorithm. You can specify PIVff on axes 1 and 2 only.	The control parameters used with the PIVff are KP, KI, KVff, KV and KIL.
No Control	Indicates that the on-board controller is to be by-passed. You can specify this option only in conjunction with the Voltage +/- 10V under Profile.	When you specify No Control, none of the control parameters are used.

The six control modes are described below:

Control mode	Description
KP	Specifies the proportional gain in a standard PID controller.
KI	Specifies the integral gain in a standard PID controller.
KD	Specifies the derivative gain in a standard PID controller.
KV	Specifies the velocity gain in a PIVff controller.
KVff	Specifies the velocity feed forward gain in a PIVff controller.
KIL	Limits the contribution from the integral term in both the PID controller and the PIVff controller.

## Setting up a profile

You can define the behavior to be expected for a given input to the Tech80 5650A block. There are four profile modes and four associated profile parameters.

**Note:** Different profiles expect inputs that are different in both type and magnitude, and mismatched inputs and profiles can cause unexpected behavior.

Profile mode	Description
Velocity	Uses a velocity input in units of revolutions per second and will accelerate the axis to that velocity at the maximum acceleration.
Trapezoid	Uses a position input in units of revolutions and tries to move the axis to that position in a trapezoidal manner. In other words, it accelerates with a constant acceleration to a fixed velocity, then moves with that velocity until it nears the destination. It then

	slows with the same magnitude on acceleration until it stops at the desired position. The Velocity and Accel parameters are used for this profile mode.
S-Curve	Uses a position input in units of revolutions and tries to move the axis to that position in an S-Curve manner. Using constant accelerations, it starts the acceleration at zero and increases it with a constant jerk (rate of change in acceleration) until it reaches the maximum acceleration. When approaching the maximum velocity, it slows the acceleration to zero at the rate of the maximum jerk.
Voltage	Uses the input as a voltage and passes that number directly to the motor output. The input can vary from -10V to 10V, with -10V causing the maximum velocity in the negative direction and 10V causing the maximum velocity in the positive direction.

Profile parameters	Description
Velocity	Indicates the maximum velocity in the Trapezoid and S-Curve profiles.
Accel	Indicates the maximum acceleration in the Velocity, Trapezoid and S-Curve profiles.
Jerk	Indicates the maximum jerk in the S-Curve profile.

## Setting the zero position

Controls whether the current position is defined as the zero position. When this option is activated, all ensuing position commands are referenced from the current position. When this option is de-activated, the current position is used “as is.”

Typically, you activate Zero Position when the motion does not return to zero. For example, suppose you de-activate Zero Position, then issue a motion sequence that starts at zero, but the axis is not at zero. Before continuing with the motion, the controller tries to go to zero, which can result in violent initial movement.

## Setting up the base address for each axis

The Base Address parameter indicates the base address the board uses for each axis. This address must match the jumper settings on the board. The base address is specified as a hexadecimal number; the leading 0x is required.

The Axis parameter indicates the axis to be controlled by the Tech80 5650A block. Each block must control a different axis.

## Setting up motor resolution

The Motor Resolution parameter indicates the counts per revolution of the motor. It may be different for each axis.

## Specifying an output type

There are two output types: DAC 16 and PWM. When selected, the DAC 16 parameter invokes the voltage output mode to the motor. The DAC16 is a 16-bit D-to-A converter that produces a +/- 10V. This is the default. The PWM parameter invokes the Pulse Width Modulated output to the motor. When you select PWM, you must make some jumper changes on the board.





# Connecting to Boards with an 8255 Chip

Keithley MetraByte PIO-24 board; Measurement Computing CIO-DIO24 and CIO-DIO24H boards; and National Instruments PC-DIO-24, PC-DIO-96, and DAQCard-1200 offer several ways to connect VisSim channels to the electrical connector terminals. These boards are capable of 24 channels of digital I/O, and can be configured as all digital inputs, all digital outputs, or byte-wise combinations of both. The DIO24 can be configured as all input channels; all output channels; or 12 input and 12 output channels.

---

## Configured as all input channels

VisSim input channel	Terminal connector channel	VisSim input channel	Terminal connector channel
0	A0	12	B4
1	A1	13	B5
2	A2	14	B6
3	A3	15	B7
4	A4	16	C0
5	A5	17	C1
6	A6	18	C2
7	A7	19	C3
8	B0	20	C4
9	B1	21	C5
10	B2	22	C6
11	B3	23	C7

---

## Configured as all output channels

VisSim output channel	Terminal connector channel	VisSim output channel	Terminal connector channel
0	A0	12	B4
1	A1	13	B5
2	A2	14	B6
3	A3	15	B7
4	A4	16	C0
5	A5	17	C1
6	A6	18	C2
7	A7	19	C3
8	B0	20	C4
9	B1	21	C5
10	B2	22	C6
11	B3	23	C7

---

## Configured as 12 inputs and 12 outputs

VisSim input channel	Terminal connector channel	VisSim output channel	Terminal connector channel
0	A0	0	B0
1	A1	1	B1
2	A2	2	B2
3	A3	3	B3
4	A4	4	B4
5	A5	5	B5
6	A6	6	B6
7	A7	7	B7
8	C0	8	C4
9	C1	9	C5
10	C2	10	C6
11	C3	11	C7

# Connecting to PCM-CIA DAS 16/12 and DAS 16/16 Boards

It is important to be aware of how to connect Measurement Computing PCM-CIA DAS 16/12 and 16/16 boards. The following table describes how the VisSim channels map to the terminal connector channels.

---

## Configured as four inputs and four outputs

VisSim input channel	Terminal connector channel	VisSim input channel	Terminal connector channel
0	DIO 0	0	DIO 4
1	DIO 1	1	DIO 5
2	DIO 2	2	DIO 6
3	DIO 3	3	DIO 7



# Installing VisSim/Real-TimePRO

The Install program that comes on your VisSim/Real-TimePRO disk installs the VisSim/Real-TimePRO program and other utility files on your hard disk.

---

## Installation requirements

VisSim/Real-TimePRO runs on personal computers using the Intel 80286 or higher processor, including the IBM Personal System/2 Series, the IBM PC AT, and 100% compatibles. To use VisSim/Real-TimePRO, your computer must have the following components:

- Visual Solutions VisSim 8.0+
- 3MB of free hard disk space

---

## Installation procedure

You use the Install program to install VisSim/Real-TimePRO on your computer for the first time or to upgrade your existing copy of VisSim/Real-TimePRO to a more recent version of the software.

At the completion of the installation, the category Real Time is added to the Blocks menu.

### To install VisSim/Real-TimePRO

This procedure assumes that you are installing from drive A to your hard disk. If you are installing from a different drive, substitute the correct drive designation in the installation procedure.

1. Insert the disk labeled VisSim/Real-TimePRO into drive A.
2. Do one of the following:
  - Click on Start and choose Run.
  - Select File from the Program Manager menu bar and choose the Run command.
  - In the Command Line box, type A:INSTALL and click on the OK button, or press ENTER.
3. Follow the on-screen instructions.



# Index

## A

Acceleration 24  
Advanced 36

## B

Boards supported by the encoder block 57  
Boards supported by the frequency block 55  
Boards supported by the PWM block 53  
Boards supported by the Tech80 5650A block 61

## C

Configured as 12 inputs and 12 outputs 68  
Configured as all input channels 67  
Configured as all output channels 67  
Configured as four inputs and four outputs 69  
Configuring a Tech80 5650A block 62  
Configuring ISA Bus I/O Boards 9  
Configuring VisSim/Real-TimePRO to recognize your I/O boards 9  
Connecting to Boards with an 8255 Chip 67  
Connecting to PCM-CIA DAS 16/12 and DAS 16/16 Boards 69  
Counter channel class 16, 19  
Current 25  
Current channel class 16

## D

Data acquisition guidelines 6  
Determining the direction of rotation value 59  
Developing Servo Control Systems 61  
Digital channel class 16, 19

## E

Edges 39

## F

Finding out an I/O board's base address 9  
Frequency 25, 40

## I

Input and output channel basics 13  
Installation procedure 71  
Installation requirements 71  
Installing and configuring I/O boards 9  
Installing VisSim/Real-TimePRO 71  
Interactive webinars 3  
Introduction 1

## M

Measuring Frequency Input 55

## N

NI-DAQmx Analog In 23  
NI-DAQmx Analog Out 45  
NI-DAQmx Counter In 38  
NI-DAQmx Counter Out 47  
NI-DAQmx Digital In 37  
NI-DAQmx Digital Out 46

## O

Options 23, 39

## P

Parameters 24, 39  
Performance degradation 7  
Period 41  
Producing Pulse Width Modulated Output 53  
Pulse Width 42

## R

Reading Encoder Input 57  
Real-time drivers 5  
Reset command 48  
Resistance 27  
Resources for learning VisSim/Real-TimePRO 3

## S

Sample diagrams 3  
Semi Period 43  
Setting Input and Output Channels with the RT Blocks 13  
Setting the analog input and output ranges 10  
Setting the base address 11  
Setting the board type and number 10

- Setting the channel settling time 11
- Setting the external mux settling time 11
- Setting the zero position 64
- Setting up a profile 63
- Setting up motor resolution 64
- Setting up the base address for each axis 64
- Sound Pressure (Microphone) 26
- Specifying a channel class 15, 19
- Specifying a channel number 14, 18
- Specifying a channel title 14, 18
- Specifying a channel type 17, 20
- Specifying a control type 63
- Specifying a mux gain for a multiplex board 15
- Specifying a sampling rate 56
- Specifying a title 58
- Specifying an output type 64
- Specifying cold junction 17
- Specifying properties of a frequency block 55
- Specifying properties of an encoder block 57
- Specifying properties of the PWM block 54
- Specifying subchannels 15
- Specifying the base address 58
- Specifying the board number 14, 18
- Specifying the channel 58
- Specifying the counter reset value 58
- Specifying the counter set mode value 58
- Specifying the signal inversion 58
- Strain 29
- Supported boards 6

## T

- Tech80 5650A block input 61
- Tech80 5650A block outputs 62
- Temperature/Built-In Sensor 30
- Temperature/RTD 28
- Temperature/Thermistor with Current Excitation 32
- Temperature/Thermistor with Voltage Excitation 33
- Temperature/Thermocouple 31
- The Basics 5
- The Real Time blocks 6
- The VisSim product family 1
- Thermocouple channel class 16
- Time Between Edges 44
- Training 3

## U

- Using a Tech80 5650A block 61
- Using the encoder block 57
- Using the File > Real Time Config command 10
- Using the frequency block 55
- Using the MCC Input block 49
- Using the MCC Output block 50
- Using the Measurement Computing Universal Driver Blocks 49

- Using the NI-DAQmx Blocks 23
- Using the PWM block 53
- Using the rt-DataIn block 13
- Using the rt-DataOut block 18

## V

- Voltage 34
- Voltage with Excitation 35
- Volts channel class 16, 20