
PXle-5413 User Manual

2025-07-22



Contents

PXle-5413 User Manual	4
PXle-5413 Overview	5
PXle-5413 Front Panel and Pinout.....	7
PXle-5413 LED Indicators.....	10
Safety Guidelines	13
EMC Standards.....	14
PXle-5413 Theory of Operation	15
PXle-5413 Block Diagram	16
PXle-5413 Product Features	17
PXle-5413 Output.....	17
Output Modes	17
Standard Function Mode	17
Arbitrary Waveform Output Mode	18
Arbitrary Sequence Mode.....	20
Frequency List Mode	22
Script Mode	23
Output Attenuation	23
Output Enable.....	24
Output Impedance.....	24
Differential Output.....	25
Analog Output Path.....	25
Fractional Resampling and Filtering	28
Frequency Hopping and Sweeping	30
Clocks	31
Phase-Locked Looping	34
Impedance Matching.....	35
Triggers and Events	37
Streaming	44
Frequency Domain Fundamentals.....	48
Filtering and Interpolation.....	50
Power On and Reset	53
Integration and System Considerations	54

Finding Examples	58
PXle-5413 Installation	60
Unpacking the Kit	60
Installing the Software	61
Installing the PXle-5413 into a Chassis	62
Performing Initial Hardware Configuration	63
Self-Calibrating the PXle-5413.....	66
Signal Connections to the PXle-5413	67
PXle-5413 Configuration	71
Configuring an Internal Sample Clock	71
Configuring a Reference Clock	71
Configuring Triggers	72
Creating a Marker Event	72
Writing a Script.....	74
Streaming Waveform Data.....	75
Configuring Independent-Channel Waveform Generation.....	82
Simulating the PXle-5413	83
Generating Waveforms	84
Cleaning the PXle-5413 System	86
PXle-5413 Fundamental Concepts	87
PXle-5413 Bandwidth and Passband Flatness.....	87
PXle-5413 Sample Rate.....	88
PXle-5413 DAC Resolution	90

PXIe-5413 User Manual

The PXIe-5413 User Manual provides detailed descriptions of the product functionality and the step by step processes for use.

Looking for Something Else?

For information not found in the User Manual for your product, such as specifications and API reference, browse ***Related Information***.

Related information:

- [PXIe-5413 Specifications](#)
- [PXIe-5413 Calibration Procedure](#)
- [NI-FGEN User Manual](#)
- [NI-FGEN LabVIEW VIs](#)
- [NI-FGEN C Function Reference](#)
- [NI-FGEN .NET Framework 4.0](#)
- [NI-FGEN .NET Framework 4.5](#)
- [NI-FGEN Python Documentation](#)
- [Software and Driver Downloads](#)
- [Release Notes](#)
- [License Setup and Activation](#)
- [Product Certifications](#)
- [Letter of Volatility](#)
- [Discussion Forums](#)
- [NI Learning Center](#)
- [NI Support](#)

PXIe-5413 Overview

The PXIe-5413 is a 20 MHz, 1- or 2-channel, 16-bit arbitrary waveform generator featuring capabilities of generating user-defined, arbitrary, and standard function waveforms, advanced synchronizing, and waveform scripting and streaming. Use the PXIe-5413 to generate signals from -12 V to +12 V and precisely generate waveforms via a fractional resampling method.

PXIe-5413 Key Features

The PXIe-5413 is a PXI waveform generator that has the following features and capabilities.

- 200 MSa/s sampling rate with digital filter enabled
- 0.00775 V_{pk-pk} to 12 V_{pk-pk} output amplitude into a 50 Ω load
- 0 MHz to 20 MHz sine wave generation
- 0 MHz to 10 MHz square wave generation
- 0 MHz to 1 MHz ramp and triangle wave generation
- Standard function, arbitrary waveform, arbitrary sequence, and frequency list output modes
- Single, continuous, stepped, and burst trigger modes when configured in frequency list, arbitrary waveform, or arbitrary sequence output modes
- Independent channels (two-channel PXIe-5413 only)
- NI-TClk synchronization

PXIe-5413 Driver Support

Determine the earliest driver version supported for your product.



Tip To optimize product performance, update to the most recent driver version.

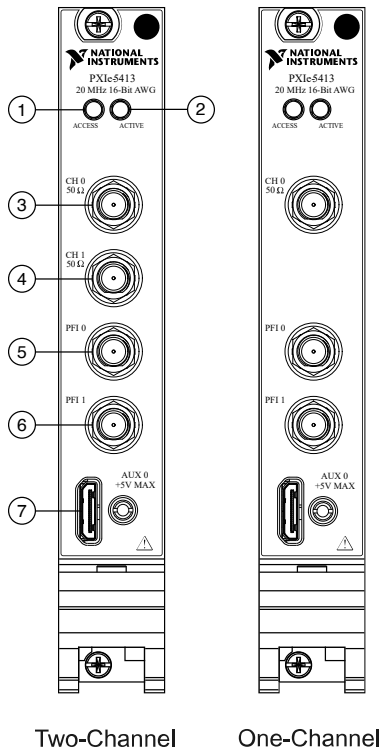
Table 1. Earliest Driver Version Support

Product	Driver Name	Earliest Version Support
PXIe-5413	NI-FGEN	17.1

PXIe-5413 Front Panel and Pinout

Understand the connectors, pinouts, and signals on the PXIe-5413 front panel.

Figure 1. PXIe-5413 Front Panel



1. Access LED
2. Active LED
3. SMA Connector: CH 0
4. SMA Connector: CH 1
5. SMA Connector: PFI 0
6. SMA Connector: PFI 1
7. MHDMR Connector: AUX 0

Table 2. SMA Connector Signal Descriptions

Signal	Access	Description
CH 0	Output	Generates waveforms from an analog output terminal.
CH 1		
PFI 0	Input/	Imports digital trigger signals and exports digital event signals.

Signal	Access	Description
PFI 1	Output	- Imported digital trigger signals can start or step through waveform generation. - Exported event signals indicate the state of the generation engine.

AUX 0 Connector Pinout

The AUX 0 MHDMMR port on the PXIe-5413 front panel, routes digital trigger and digital event signals with eight bidirectional PFI lines. The port also provides a +3.3 V power source.

AUX 0 also provides +24 V power for supported accessories.



Notice The AUX 0 connector accepts a standard, third-party HDMI™ type C cable. The specified performance of the AUX 0 connector is not guaranteed if you use a third-party HDMI cable. Use NI cable assembly SHH19-MH19-AUX for all AUX 0 connections. Do not connect the AUX 0 port on the PXIe-5413 to the HDMI port of another instrument. NI is not liable for any damage resulting from such signal connections.

Figure 2. AUX 0 Connector Pinout

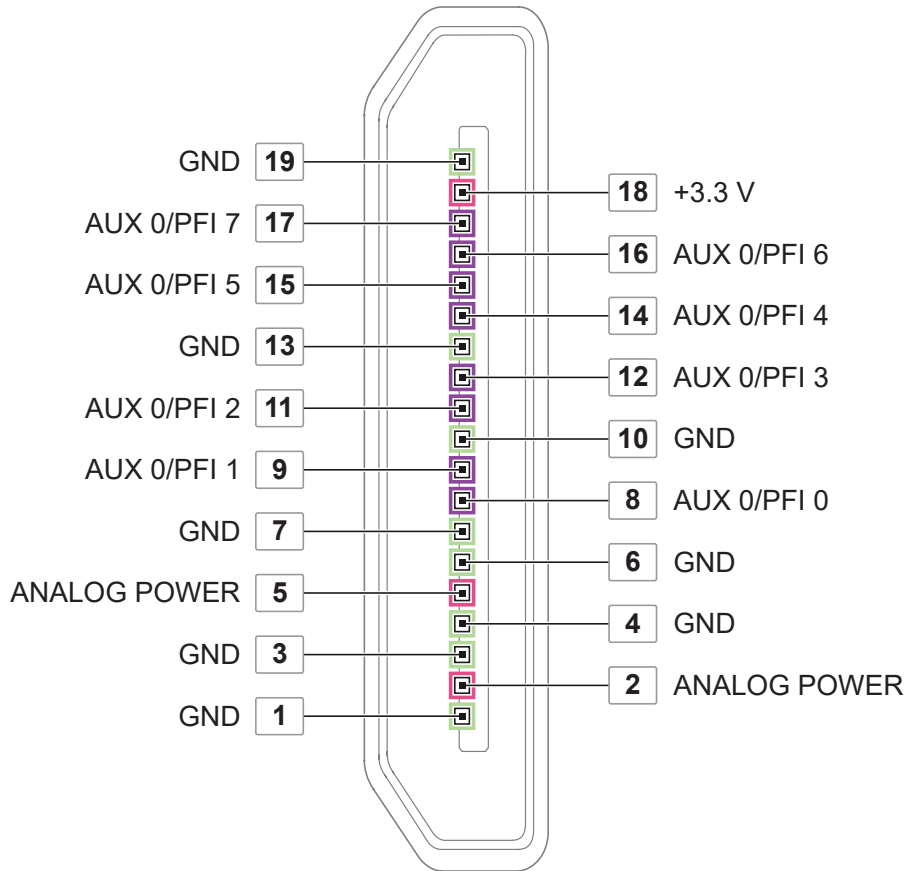


Table 3. Signal Descriptions

Signal Name	Description
GND	Ground reference for signals
AUX 0/PFI <0 ... 7>	Bidirectional PFI line
+3.3 V	+3.3 V power output (200 mA maximum)

PXle-5413 LED Indicators

The PXle-5413 features an Access LED and Active LED.

Access LED

The Access LED, located on the module front panel, indicates module power and access.

The following table lists the Access LED states.

Table 4. ACCESS LED Indicator Status

Status Indicator	Indication
No color (off)	The module is not powered.
Amber	The module is being accessed.
Green	The module is powered and ready to be programmed.

Why Is the Access LED Off When the Chassis Is On?

The LEDs may not light until the module has been configured in Hardware Configuration Utility or MAX. Before proceeding, verify that the PXle-5413 appears in Hardware Configuration Utility or MAX.

If the module appears in Hardware Configuration Utility or MAX but the Access LED fails to light after you power on the chassis, a problem may exist with the chassis power rails, a hardware module, or the LED.



Notice Apply external signals only when the PXle-5413 is powered on. Applying external signals while the module is powered off may cause damage.

1. Disconnect any signals from the module front panel.
2. Power off the chassis.

- Remove the module from the chassis and inspect it for damage.



Notice Do not reinstall a damaged module.

- Install the module in a different, supported slot within the same PXI chassis.
- Power on the chassis.



Note If you are using a PC with a module for PXI remote control system, power on the chassis before powering on the computer.

- Verify that the module appears in Hardware Configuration Utility or MAX.
- Call the niFgen Reset Device VI or niFgen_ResetDevice function to reset the module and perform a self-test.

You can also perform a self-test in Hardware Configuration Utility or MAX.

Active LED

The Active LED, located on the module front panel, indicates the module output channel state.

The following table lists the Active LED states.

Table 5. Active LED Indicator Status

Status Indicator	Indication
(Off)	The module is not generating waveforms.
Amber	The module is armed and waiting for a trigger.
Green	The module has received a trigger and is generating a waveform.
Red	The module has detected an error.

Why Is the Active LED Red?

The red Active LED indicates that the instrument is in error state. The instrument will not operate until the error is cleared and/or the module is reset.



Note NI-FGEN must access the PXIe-5413 to determine the cause of the

error.

Possible Error	Solution
The instrument has detected an unlocked condition on a PLL that might result in incorrect data.  Note A PLL that is unlocked while in reset does not show an error.	To clear this error, call the niFgen Reset Device VI or niFgen_ResetDevice function to reset the instrument.
The instrument has been disabled because it exceeded its overall power limit.	
The instrument has been disabled because it exceeded its overall temperature limit.	To re-enable the instrument, cool the instrument to an acceptable range and resolve the environmental condition that caused the shutdown. To reset the instrument, call the niFgen Reset Device VI or niFgen_ResetDevice function, or power cycle the instrument.

Related information:

- [Maintain Forced-Air Cooling Note to Users](#)

Safety Guidelines



Caution Observe all instructions and cautions in the user documentation. Using the product in a manner not specified can damage the product and compromise the built-in safety protection.



Attention Suivez toutes les instructions et respectez toutes les mises en garde de la documentation d'utilisation. L'utilisation du produit de toute autre façon que celle spécifiée risque de l'endommager et de compromettre la protection de sécurité intégrée.

EMC Standards

This product meets the requirements of the following EMC standards for electrical equipment for measurement, control, and laboratory use:

- EN 61326-1 (IEC 61326-1): Class A emissions; Basic immunity
- EN 55011 (CISPR 11): Group 1, Class A emissions
- AS/NZS CISPR 11: Group 1, Class A emissions



Note Group 1 equipment is any industrial, scientific, or medical equipment that does not intentionally generate radio frequency energy for the treatment of material or inspection/analysis purposes.



Note In Europe, Australia, and New Zealand (per CISPR 11) Class A equipment is intended for use in non-residential locations.

PXIe-5413 Theory of Operation

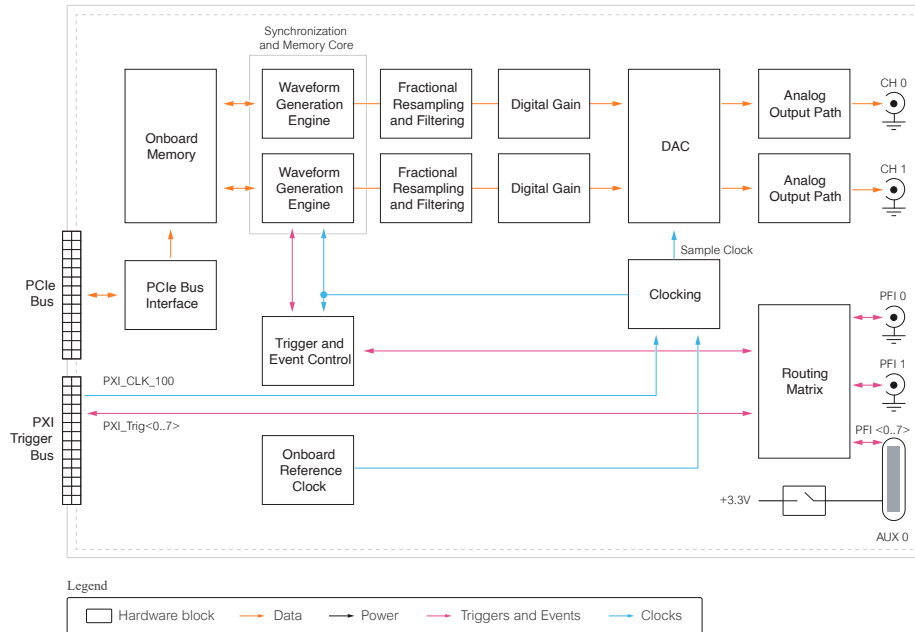
The PXIe-5413 generates analog output signals from digital waveform data at the CH connectors and routes triggers, events, and clock signals.

- **Onboard Memory**—Stores the waveform data and generation instructions that you load into the PXIe-5413.
- **Waveform Generation Engine**—Retrieves the waveform data and instructions from the onboard memory and triggers from trigger and event control using the sample clock.
- **Fractional Resampling and Filtering**—Resamples (digital signal processing) and filters images.
- **Digital Gain**—Applies digital gain to the signal.
- **DAC**—Converts 16-bit digital waveform data.
- **Analog Output Path**—Filters, amplifies, and attenuates the waveform data retrieved from the DAC.
- **Clocking**—Creates and distributes the sample clock.
- **Onboard Reference Clock**—Provides the default reference clock source.
- **Trigger and Event Control**—Sends triggers and events to the waveform generation engine and routing matrix.
- **Routing Matrix**—Routes triggers and events between PXI trigger lines and PFI terminals.

PXIe-5413 Block Diagram

The following diagram illustrates the design of the PXIe-5413.

Figure 3. PXIe-5413 Block Diagram



PXIe-5413 Product Features

Understanding the PXIe-5413 features can help you use the instrument to meet your testing needs.

PXIe-5413 Output

Refer to the following topics for details about the PXIe-5413 output.

Output Modes

The output mode of your signal generator determines the type of waveforms your signal generator produces. To select an output mode, set the `Output Mode` parameter of the `niFgen Configure Output Mode VI` or the `outputMode` parameter of the `niFgen_ConfigureOutputMode` function.

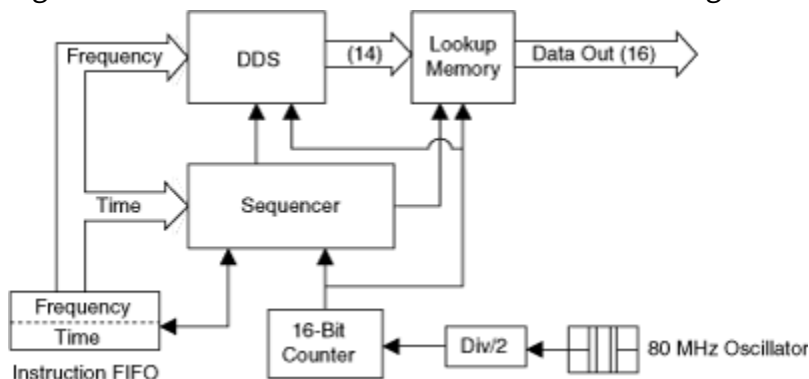
Standard Function Mode

Standard function output mode is used to generate standard function waveforms, including sine, square, and triangle.

Direct Digital Synthesis

The standard function output mode is implemented through direct digital synthesis (DDS) on the PXIe-5413.

Figure 4. DDS-Based Waveform Generation Block Diagram



You can use the lookup memory in standard function output mode only. You can store one cycle of a repetitive waveform—a sine, triangular, square, or arbitrary wave—in the lookup memory. Then, you can change the frequency of that waveform by sending just one instruction. You can use standard function output mode for very fine frequency resolution function generation. Because standard function output mode uses an accumulator, waveform generation loops back to the beginning of the lookup memory after passing through the end of the lookup memory.

In standard function output mode, each stage is made up of two instructions: the frequency and the time.

- **Frequency instruction**—Specifies the frequency of the waveform generation.
- **Time instruction**—Specifies the duration when the frequency is generated.



Note You cannot specify the number of iterations for a waveform to generate in Standard Function output mode.

Arbitrary Waveform Output Mode

For the PXIe-5413, arbitrary waveform output mode generates waveforms from user-created or user-provided waveform arrays of numeric data.

The waveform arrays are downloaded to the onboard memory on the instrument. Arbitrary waveform output mode also uses memory to store the instructions on generating waveform sequences in the waveform memory.



Note You cannot use the lookup memory in arbitrary waveform output mode.

Marker Output Signal

NI signal generators can generate **markers** at specific points within a waveform. A **marker** is a digital pulse that can be used as a trigger to control the timing of other devices in your application. You can route this signal to connectors on the PXIe-5413 front panel or to PXI trigger lines. You can place a marker in every sequence segment.

You can specify a marker by giving an offset count, in number of samples, from the start of the waveform buffer for each segment. If the waveform loops multiple times in

a segment, the marker generates each time the waveform loops at the configured sample position.

If you need to generate a marker only once in a segment in which the waveform loops a number of times, break the segment up into multiple segments. For example, if you need to generate a marker on only the first iteration of a sine waveform that loops 100 times, create a segment that loops only once and generates the sine waveform with the marker at a specific sample position. Create a second segment that generates the sine waveform with no marker and loops 99 times.

This technique can be used in different combinations of segments to generate one marker every 10 of the sine waveform by creating a segment containing a marker and one loop followed by a segment with no markers that loops nine times. Repeat these two segments for as long as you need to repeat the cycle.

Minimum Waveform Size and Quantum

The memory architecture of the NI signal generators imposes certain requirements on the waveform size and quantum. If these requirements are not met, NI-FGEN returns an error. The specific values for minimum waveform size and quantum depend on the specific NI signal generator.



Note Refer to *PXIe-5413 Specifications* for minimum waveform size and quantum values.

Minimum Waveform Size

Every waveform downloaded to the instrument memory must be at least a minimum size in terms of the number of samples.

Quantum

The size in samples of the waveform downloaded to the instrument memory must be an integer multiple of a certain number or quantum.

For example, if the minimum waveform size for a particular signal generator is 256 samples, the quantum is eight samples, and you request to load a waveform of 255 samples, NI-FGEN returns an error because the waveform size is too small. If you

request to load a waveform of 257 samples, NI-FGEN also returns an error because even though the size is larger than the minimum waveform size, the waveform is not an integer multiple of the quantum size (8).

Waveform sizes that meet the restrictions for this example include 256 sample, 264 samples, 272 samples, 280 samples, and others within the actual instrument memory size limit.

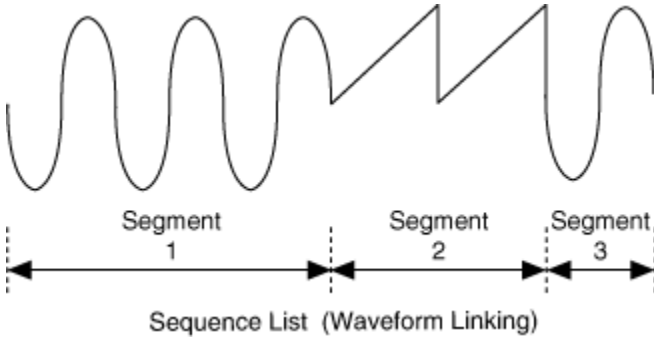
Arbitrary Sequence Mode

Arbitrary sequence output mode allows you to load multiple waveforms in the onboard memory of the signal generator.

A finite number of samples make a waveform. To generate these downloaded waveforms in a specific order, you must prepare a sequence, which contains a number of segments in a specific order. Each segment specifies a downloaded waveform, a number of loops to repeat the selected waveform, and a numeric offset in which a marker is generated by the instrument. The timing and behavior of the generation of a waveform sequence is dependent on the trigger mode selected.

Table 6. Waveform and Segment

Waveform or Segment Type	Waveform or Segment Shape	Description
Waveform A	 Waveform A	A single cycle of a sine wave that is downloaded to onboard memory
Waveform B	 Waveform B	A single cycle of a ramp waveform that is downloaded to onboard memory

Waveform or Segment Type	Waveform or Segment Shape	Description
Waveform Segment 1	 Waveform Segment 1 (Loops = 3)	A segment created using Waveform A that repeats or loops three times
Waveform Segment 2	 Waveform Segment 2 (Loops = 2)	A segment created using Waveform B that loops twice
Waveform Segment 3	 Waveform Segment 3 (Loops=1)	A segment created using Waveform A that loops only once
Sequence List or Waveform Linking	 Sequence List (Waveform Linking)	The waveforms are linked in a sequence. The concept of using a sequence to generate waveforms is referred to as waveform sequencing or linking and looping waveforms .

Segment Components

You create a sequence segment by segment. Each segment is made up of the following four components.

- **Waveform Handle**—Specifies the downloaded waveform to be accessed by the segment. A waveform handle is returned for each waveform that is downloaded to the onboard memory. Refer to the niFgen Create Waveform or niFgen Allocate Waveform VIs, or the niFgen_CreateArbWaveform or niFgen_AllocateWaveform functions for more information about downloading waveforms.
- **Sample Count**—Specifies how many samples of a downloaded waveform the

segment uses. The sample count is not necessarily the actual size of the downloaded waveform. If the sample count is less than the actual size of the downloaded waveform, only a part of that waveform is used for that segment, starting with the first sample of the waveform. If the count is more than the actual size of that waveform, NI-FGEN returns an error. If the sample count is set to zero, NI-FGEN automatically uses the actual size of that waveform.

- **Waveform Loops**—Specifies the number of times that the waveform, or portion indicated by sample count of the waveform loops. The maximum number of loops is 16,777,215.
- **Marker Offset**—Specifies where the marker generates within that waveform. The offset is referenced to the beginning of the waveform, with sample 0 being the first sample of the waveform. Refer to **Marker Events** for more information.

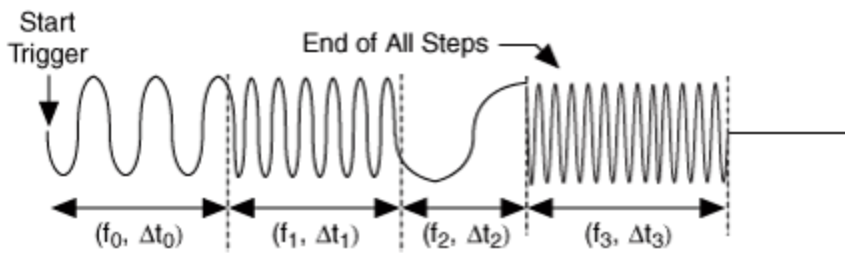
Refer to the niFgen Create Arbitrary Sequence or niFgen Create Advanced Arb Sequence VIs or the niFgen_CreateArbSequence or niFgen_CreateAdvancedArbSequence functions for more information about creating and configuring arbitrary sequences.

Frequency List Mode

In frequency list output mode, the instrument generates a standard function using a list of frequencies you define. A **frequency list** is composed of steps, each one with a frequency/duration pair. Frequency lists are commonly used for frequency sweeps and frequency-shift-keying (FSK) applications.

The following example shows a frequency list and the generated waveform when single trigger mode is selected. The trigger mode affects the timing and behavior of the waveform generation.

Step	Frequency	Duration
0	3.0 Hz	1.0 s
1	6.0 Hz	0.9 s
2	0.5 Hz	0.75 s
3	9.0 Hz	1.1 s



Script Mode

Script mode allows you to use scripting to link and loop multiple waveforms in complex combinations using a script. A **script** is a series of instructions that indicates how waveforms saved in the onboard memory should be sent to the DUT. The script can specify the order of the generated waveforms, the number of times they are generated, and the triggers and markers associated with the generation.

Output Attenuation

Output attenuation is a method of controlling the output voltage level of the signal being generated. NI signal generators typically generate signals with a digital-to-analog converter (DAC) that has an output voltage range from ± 1.0 V to ± 5.0 V with a number of bits of resolution. This signal is then applied to switchable amplifiers and attenuators to control the output voltage range of the signal source.

By attenuating the DAC output signal, you keep the dynamic range of the DAC; that is, you do not lose any bits from the digital representation of the signal because the attenuation is done after the DAC and not before it.

For example, if a DAC with a range from -5.0 V to $+5.0$ V and a resolution of 12 bits with each bit corresponding to 2.44 mV $[(+5.0 - (-5.0)) / 2^{12}]$ does not use output attenuation, and the desired signal is $+2.0$ V_{pk-pk} (-1.0 V to $+1.0$ V), waveform values can be generated with the DAC that only use $1/5$ of the DAC range. The resolution of each digital bit is still 2.44 mV.

However, if the same DAC uses the output attenuation, the full range of the DAC generates the signal, creating the signal at the full 10.0 V_{pk-pk}. The value of each digital bit is still the original 2.44 mV. The signal is applied to an attenuator, which reduces the voltage level by a factor of 5 to 2 V_{pk-pk}. The attenuator also reduces the value of each

bit, which results in an effective bit value of 0.488 mV at the analog output connector. The attenuator allows the use of the full range of the DAC, and reduces the effective value of each bit corresponding to the degree of attenuation.

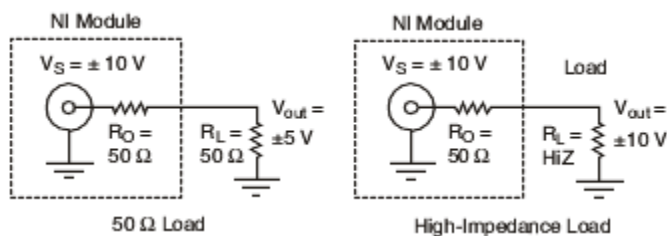
Output Enable

You can switch off the waveform generation at the output connector by controlling the output enable relay on the signal generator. When the output enable relay is disabled, the output signal level goes to ground level and waveform generation continues.

Output Impedance

The PXIe-5413 has an output impedance of 50 Ω .

If the load impedance matches the output impedance, the voltage at the signal output connector is at the needed level. The voltage at the signal output connector varies with load output impedance, up to doubling the voltage for a high impedance load, as shown in the following figure.



In the previous figure, the required output voltage is $\pm 5\text{ V}$. The NI signal generator internal voltage source generates twice this voltage, $\pm 10\text{ V}$, expecting the voltage to be halved due to the voltage divider formed by R_O and R_L . The relationship between the output voltage and the load can be calculated using the following equation:

$$V_{out} = \left[R_L \div (R_L + R_O) \right] \times [V_S]$$

where

- V_{out} = the voltage level delivered to R_L
- R_L = the load impedance in ohms
- R_O = the output impedance on the NI signal generator

- V_S = the voltage level generated by the source previous to R_O

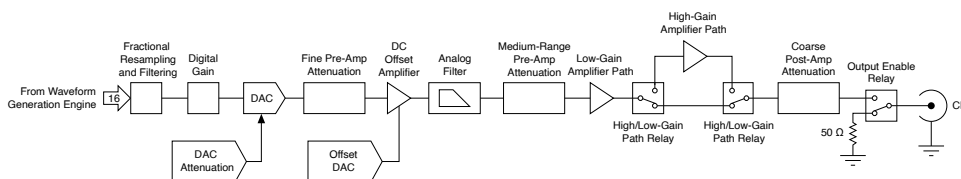
By default, $R_O = 50\ \Omega$. With some instruments you can enter the value of R_L , and the instrument changes V_S accordingly to deliver the requested V_{out} .

Differential Output

An NI signal generator that is configured for differential output generates waveforms using two complementary signals. Each signal is generated at a different differential output connector. When the two signals are compared at a receiver that has good rejection, any common-mode noise, that is, noise common between the two output terminals, and even-order distortions, for example, 2nd, 4th, 6th, ... harmonics, affect whether the receiving instrument can detect an accurate signal.

Analog Output Path

The two-channel PXIe-5413 supports independent-channel waveform generation.



The following processes outline the generation of analog waveforms with the PXIe-5413:

1. The 16-bit digital waveform data from the waveform generation engine is passed through fractional resampling and filtering.



Note When the PXIe-5413 is configured to standard function output mode, the signal bypasses fractional resampling and filtering.

2. The signal passes through a digital gain circuit.

Digital gain multiplies waveform data by a factor you specify with the `Digital Gain` property or the `NIFGEN_ATTR_DIGITAL_GAIN` attribute before converting the data to an analog signal in the DAC.

Digital gain can be changed during waveform generation without the glitches caused by switching that are common when changing analog gain. However, the output resolution of the DAC is a function of digital gain, meaning that only analog gain makes full use of the resolution of the DAC.

3. The signal passes through the 800 MSa/s DAC.
4. The signal passes through fine preamplifier attenuation: a set of selectable solid-state attenuators that provide 0 dB to 2.4 dB of signal attenuation in sixteen 0.16 dB increments. The final 0.16 dB increment uses DAC-code scaling to configure the exact amplitude based on the gain setting.
5. The signal passes through a DC offset amplifier that adds the specified DC offset voltage.

You can adjust the amount of DC offset added to the signal, up to one-half of the value of the gain setting, by using either the `Arbitrary Waveform Offset` property or `DC Offset` property, or the `NIFGEN_ATTR_ARB_OFFSET` or `NIFGEN_ATTR_FUNC_DC_OFFSET` attributes.

6. The signal passes through an enabled 80 MHz time-domain, Bessel analog filter. The enabled analog filter increases the delay from the time at which the PXle-5413 receives a trigger to the time at which the analog output signal is generated.



Note When the PXle-5413 is configured for sine wave generation in standard function output mode, the signal bypasses the time-domain, Bessel filter.

7. The signal passes through medium-range preamplifier attenuation: a set of selectable solid-state attenuators that provide 0 dB to 10.23 dB of attenuation in five 2.56 dB increments.

You can adjust the amount of attenuation by configuring the `Arbitrary Waveform Gain` property or `Amplitude` property, or the `NIFGEN_ATTR_ARB_GAIN` or `NIFGEN_ATTR_FUNC_AMPLITUDE` attributes. NI-FGEN calculates and sets the correct amount of attenuation required, corresponding to the gain setting.

8. The signal continues through either the low-gain amplifier path or high-gain amplifier path.

- The low-gain amplifier path is used for waveforms that have all output voltages equal to or less than ± 2.75 V into 50 Ω loads. The low-gain amplifier has a fixed gain.
- The high-gain amplifier path is used for waveform output voltages greater than ± 2.75 V into 50 Ω loads. The high-gain amplifier has a fixed gain and enables maximum peak-to-peak signal amplitude (V_{pk-pk}).

By default, NI-FGEN automatically selects the amplifier path the signal passes through based on the gain or amplitude setting.

Load Impedance	Maximum Gain ($V_{pk-center}$)	Maximum Amplitude (V_{pk-pk})
50 Ω	6	12
1 M Ω	12	24

Alternatively, you can use the `Analog Path` property or the `NIFGEN_ATTR_ANALOG_PATH` attribute to set the signal path to remain constant regardless of the gain setting for applications requiring only one of the amplifier paths. Configuring a constant signal path through the low-gain amplifier path regardless of the gain setting affects the maximum output value allowable for that particular gain setting: The maximum gain setting for signals configured to use the low-gain amplifier path is 2.75.

9. The signal passes through coarse postamplifier attenuation: a set of two passive attenuators that provide 0 dB to 38.37 dB of attenuation in four 12.79 dB increments.

You can adjust the amount of attenuation by configuring either the `Arbitrary Waveform Gain` property or `Amplitude` property, or the `NIFGEN_ATTR_ARB_GAIN` attribute or `NIFGEN_ATTR_FUNC_AMPLITUDE` attribute. NI-FGEN calculates and sets the correct amount of attenuation required, corresponding to the gain or amplitude settings.

At this point, the signal has an applied maximum attenuation of 51 dB. After this point, the signal will continue to digitally attenuate.

10. The signal passes through the output enable relay.

- When the relay is disabled, ground is connected to the output through a 50 Ω resistor. Waveform generation continues while the output enable relay is disabled.
- When the relay is enabled, the analog waveform is generated at the analog output connector.

You can enable or disable analog output using `niFgen Output Enable` or `niFgen_ConfigureOutputEnabled`.



Note When you change a setting that results in a relay switch, the bouncing of electromechanical relays on the PXle-5413 distorts the output signal for about 10 ms.

Fractional Resampling and Filtering

The 16-bit digital waveform data from the waveform generation engine can pass through different resampling digital signal processing (DSP) paths when digital filter is enabled or disabled.

To select the path for the signal to pass through, you can enable or disable the digital filter using `niFgen Configure Digital Filter`, `niFgen_EnableDigitalFilter`, or `niFgen_DisableDigitalFilter`. By default, the digital filter is enabled.

When the PXle-5413 is configured to standard function output mode, the signal bypasses fractional resampling and filtering regardless of the digital filter setting.

Digital Filter Disabled

With the digital filter disabled, the waveform output includes an update step at the sample rate you specify with `niFgen Set Sample Rate` or the `Sample Rate` property. As a result, the waveform output includes images of the original waveform at multiples of the sample rate.

Analog filtering later in the signal path causes the images to roll-off beyond 80 MHz.

The following figures show 2.5 MHz sine waveform generation at 50 MS/s with the digital filter disabled.

Figure 5. Time Domain with Digital Filter Disabled

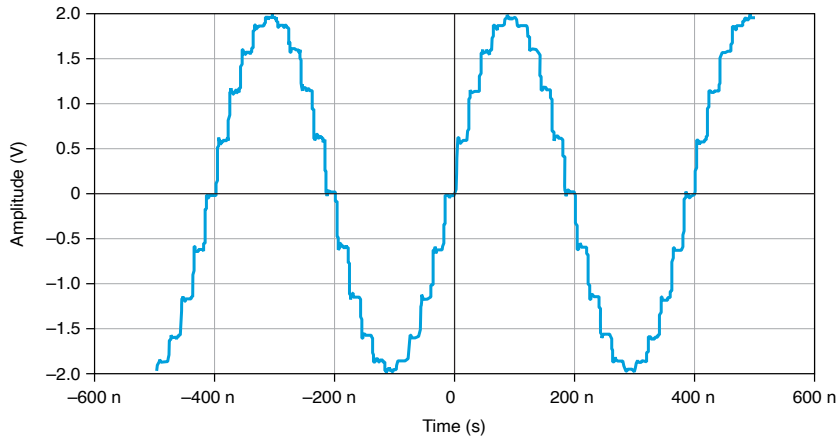
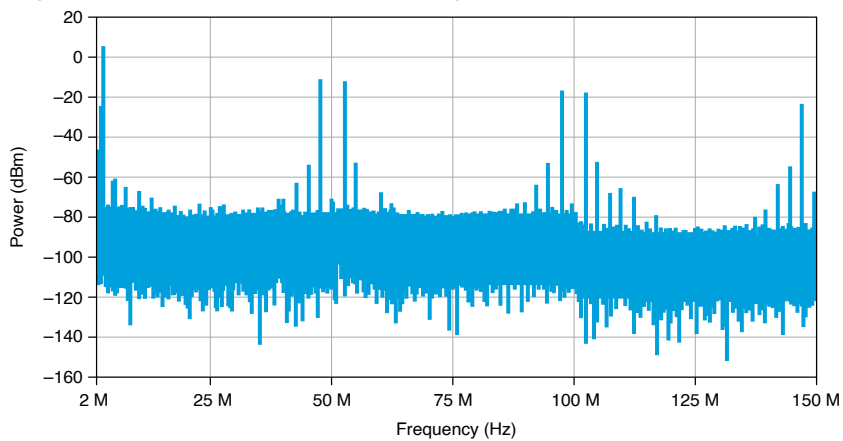


Figure 6. Power Spectrum with Digital Filter Disabled



Digital Filter Enabled

With the digital filter enabled, images at multiples of the sample rate are removed from the waveform output.

The following figures show 2.5 MHz sine waveform generation at 50 MS/s with the digital filter enabled.

Figure 7. Time Domain with Digital Filter Enabled

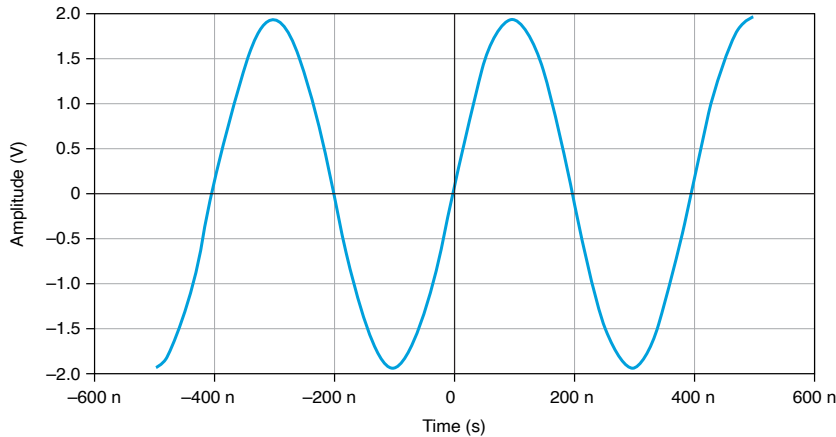
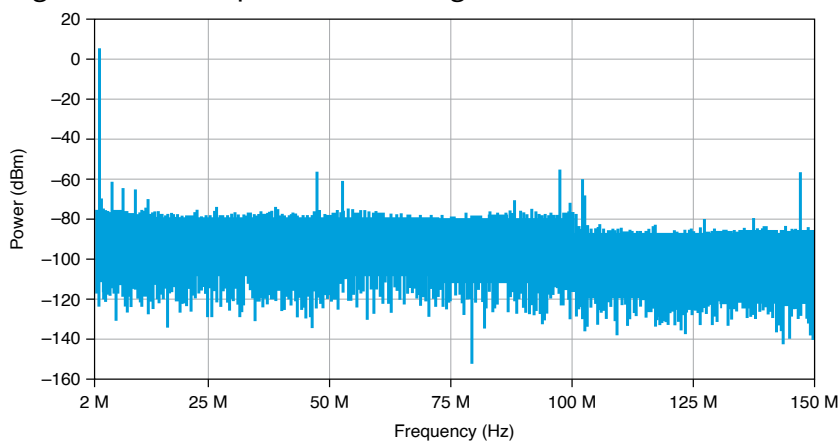


Figure 8. Power Spectrum with Digital Filter Enabled



Frequency Hopping and Sweeping

You can define a staging list for performing frequency hops and sweeps. The entire staging list uses the same waveform loaded into the lookup memory. All stages are phase-continuous and differ only with the frequency generated.

Frequency Sweeping

A **frequency sweep** is the continuous generation of a single waveform with a frequency that changes in a linear way.

The following are the basic elements used to control the generation of a frequency sweep:

- **Start frequency**—Specifies the first frequency generated in the frequency sweep.
- **End frequency**—Specifies the last frequency generated in the frequency sweep.

- **Number of frequency steps**—Specifies the number of segments into which a waveform is divided. Each frequency step corresponds to a particular frequency.
- **Frequency step duration**—Specifies the amount of time the waveform is generated at a particular frequency.

You can use these elements to programmatically create a frequency sweep.

In frequency list output mode, you can use the Fgen Sweep Generator example in LabVIEW or LabWindows/CVI.

A simple example of a frequency sweep is a chirp waveform—a sine wave produced with a linear sweep of frequency. Refer to the ***NI Analog Waveform Editor Help*** for information about creating and configuring a chirp waveform using the AWE.

Frequency Hopping

A ***frequency hop*** is applied in an order defined by the user, which is different from a ***frequency sweep*** that is applied in a linear succession. You can use frequency list, arbitrary sequence, or script output mode to implement frequency hopping.

Related information:

- [NI Analog Waveform Editor Help](#)

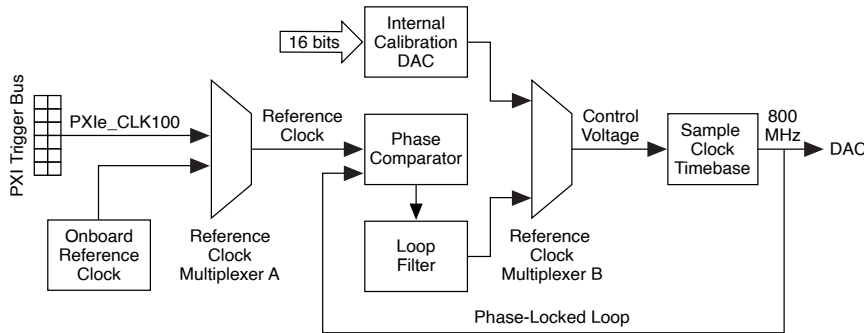
Clocks

Clocking is used to control the rate at which signals are generated.

An internal voltage-controlled crystal oscillator (VCXO) behaves as a sample clock timebase for the signal generator. This sample clock timebase in the PXIe-5413 is either configured or replaced by using one of the following methods:

- Internal sample clock
- Reference clock

Figure 9. PXle-5413 Clocking



The PXle-5413 supports the following clocking options:

- None, OnboardRefClk, OnboardReferenceClk, or OnboardReferenceClock
- PXI_Clk or PXle_Clk100

Internal Sample Clock

The internal sample clock timebase is used to determine clocking rates. In arbitrary waveform mode, you can control this timebase by using high-resolution mode in the PXle-5413.

The sample clock signals when the DAC converts the digital waveform values to an analog voltage. The PXle-5413 derives a sample clock from its main internal timing source, the onboard sample clock timebase. The PXle-5413 uses a high-precision 100 MHz voltage-controlled crystal oscillator (VCXO) clock source for the sample clock timebase.

The sample clock timebase frequency is tuned by an internal calibration DAC control voltage when the `Reference Clock Source` property or the `NIFGEN_ATTR_REFERENCE_CLOCK_SOURCE` attribute is set to `None`. The internal calibration DAC, which is calibrated at the factory and which you also can calibrate, tunes the sample clock timebase to maintain a high quality frequency source.

The PXle-5413 supports high-resolution and automatic clock mode and does not support external sample clock sources.

Clock Mode

The clock mode is applicable only when using an internal sample clock. The clock mode determines the method of deriving the sample clock from the sample clock

timebase—the main timing component of the instrument. You can set the clock mode with the `Clock Mode` property or the `NIFGEN_ATTR_CLOCK_MODE` attribute.

High-Resolution Clock Mode

High-resolution clocking allows you to set the sample clock frequency to any value from zero to the instrument sample clock timebase frequency with a very fine resolution typically in the millihertz or microhertz range. This mode is useful for applications that require a precise clock source.

Automatic Clock Mode

In automatic clock mode, NI-FGEN switches between divide-down and the high-resolution clock modes depending on the sample rate configured with the `Sample Rate` property or the `NIFGEN_ATTR_ARB_SAMPLE_RATE` attribute. Because the PXIe-5413 does not support divide-down, NI-FGEN always selects high-resolution in automatic clock mode.

Reference Clock

An external device is used as a reference clock in the phase-locked loop (PLL) of the signal generator. This causes the reference clock to phase-lock to the sample clock timebase so that the frequency stability and accuracy of the sample clock timebase matches that of the reference clock.

The onboard reference clock is a a temperature-compensated crystal oscillator (TCXO). The PXIe-5413 uses either the onboard reference clock or the PXIe_CLK100 backplane line as the reference clock source. The source provides the control voltage that tunes the VCXO of the sample clock timebase to match the frequency stability and accuracy of the reference clock source for internal clock update sources using a phase-locked loop (PLL).

The following steps describe a complete PLL instance.

1. The phase comparator compares the selected reference clock to the 100 MHz clock of the sample clock timebase to begin the PLL.
2. A control voltage proportional to the phase difference between the two clocks is developed and used to tune the sample clock timebase into alignment with the

reference clock.

3. The sample clock timebase output is routed back to the phase comparator, and the loop is closed.

You can synchronize the PXle-5413 with other instruments using PXle_CLK100 as the reference clock source and NI-TClk.

Related concepts:

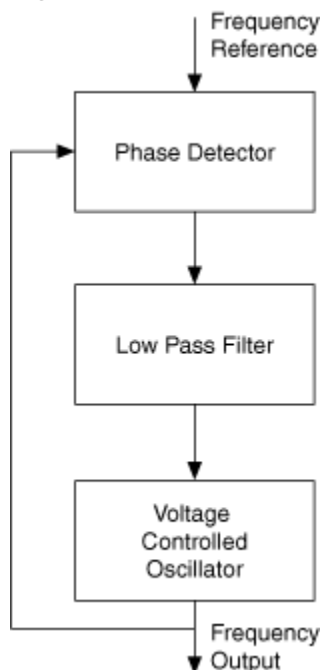
- [Synchronization](#)

Phase-Locked Loop

A **phase-locked loop (PLL)** is a circuit that adjusts a main clock to synchronize to a reference clock. The frequency stability of the sample clock timebase matches that of the reference clock when the two are phase-locked. Phase locking also synchronizes clocks of multiple instruments that are phase-locked to the same reference clock.

The following figure shows a block diagram of a basic PLL.

Figure 10. Basic PLL Block Diagram



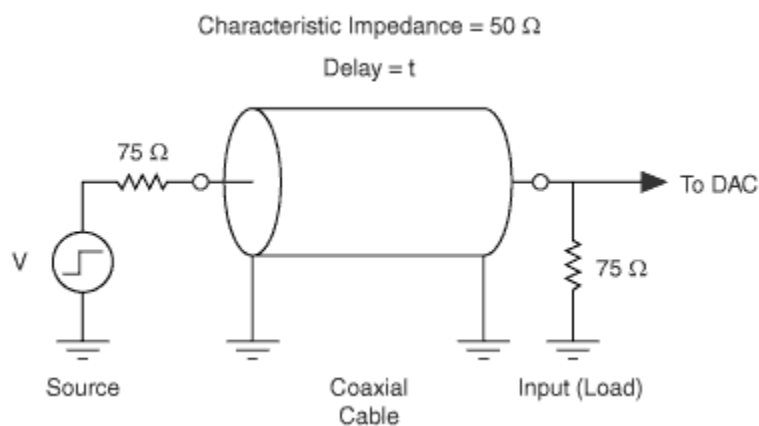
The operation of this circuit is typical of all PLLs. A PLL is a feedback control system that controls the phase of a voltage-controlled oscillator (VCO). The frequency

reference signal is applied to a phase detector. The output of the VCO connects to the other input. Normally the frequencies of both signals are almost the same. The output of the phase detector has a voltage proportional to the phase difference between the two input signals. The loop filter receives this signal from the phase detector. The loop filter determines the dynamic characteristics of the PLL.

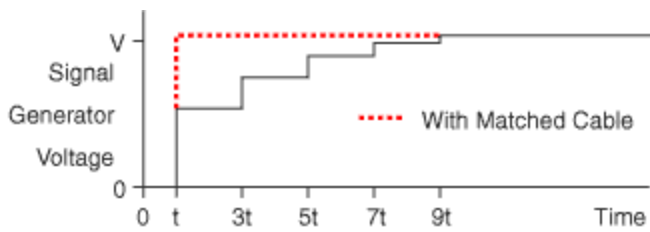
Impedance Matching

When broadband signals are carried on transmission lines of any significant length, care must be taken that the transmission line is matched to its terminations. The source and load impedances should equal the characteristic impedance of the transmission line, as this minimizes signal reflections. The presence of impedance discontinuities or mismatches degrade the amplitude and phase accuracy, as well as the temporal fidelity, of waveforms generated with a signal generator.

The following figure shows one of the most common mismatch errors encountered in such measurements.



In this example, selectable source impedances are provided at the signal generator outputs to accommodate the most popular coaxial cable characteristic impedances: 50 Ω and 75 Ω. The following figure shows what happens when, as in this example, a coaxial cable of the wrong characteristic impedance (50 Ω) is used with 75 Ω source and load impedances.



The pulse encounters impedance mismatches at each end of the cable, causing the pulse to be partially reflected. The reflected pulse traverses the cable back and forth numerous times, diminishing at each end by the reflection coefficient, Γ .

$$\Gamma = \frac{V_r}{V_i} = \frac{Z_t - Z_0}{Z_t + Z_0}$$

where

- V_r = reflected voltage
- V_i = incident voltage
- Z_t = terminating impedance
- Z_0 = characteristic impedance

The resulting voltage waveform is distorted by the asymptotic decay of the reflected pulse as shown, exaggerated for visual effect. Impedance discontinuities of smaller magnitude and/or duration have correspondingly smaller effects. Also displayed is the waveform that results when a cable of matched impedance (75 Ω) is used.

Mismatch Uncertainty

Impedance matching is also important for preserving the absolute delivered power from an instrument. The accuracy with which power can be delivered is limited by mismatch error. The mismatch error in a z_0 system can be shown to be bounded by:

$$\frac{(1 - |\Gamma_L|^2)}{(1 + |\Gamma_L| + |\Gamma_G|)^2} \leq \text{mismatch error} \leq \frac{(1 - |\Gamma_L|^2)}{(1 - |\Gamma_L| + |\Gamma_G|)^2}$$

where

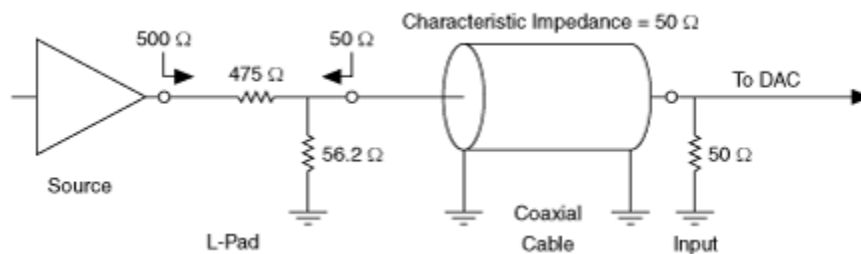
- Γ_L = load reflection coefficient

- Γ_G = generator reflection coefficient

The denominator term represents mismatch uncertainty, which is a fundamental limit to the power transfer accuracy that can be achieved across a mismatched junction.

Resistive Matching

Signal generators with low/high-source impedance can be matched with a resistor placed in series (shunt) such that the total source impedance (admittance) is matched to the cable characteristic impedance (admittance). Signal generators that are not capable of driving the cable impedance directly can be coupled through a matching L-pad. In this case, the signal generator sees an approximately 500 Ω load, while the source impedance presented to the cable is 50 Ω , as the following figure shows.



High-frequency components and layout techniques should be used throughout to minimize parasitic effects.

Triggers and Events

You can use triggers and events to coordinate the operation of multiple channels and instruments.

Related tasks:

- [Configuring Triggers](#)

Related reference:

- [Creating a Marker Event](#)

Triggers

Triggers are signals that cause the signal generator to perform an action such as starting or stopping a generation operation. Triggers can be internal (software-generated) or external. External digital triggers can be several different types. External triggers can be re-exported and, along with events, can allow you to synchronize the hardware operation with external circuitry or other NI devices. When triggering the PXle-5413 in standard function, arbitrary waveform, arbitrary sequence, script, or frequency list mode, you can select the trigger type, the trigger source, and the trigger mode that you want to use.

Events

An **event** is a signal generated by the NI device at a device state. Typically, events are configured to indicate when a specific hardware condition has been met.

Event Output Behaviors

Events can have one of the following output behaviors:

- **Toggle**—Each instance of the event toggles between high and low. You can set the initial state of the event.
- **Pulse**—Each event triggers a pulse for a specified period of time.
- **Level**—While the event is active, it shifts high or low depending on the active state you specify.

Event Status

Events can return their status in two ways.

- **Live**—Returns the current state of the event.
- **Latched**—Returns whether the event has ever been active.

Refer to the following table for the PXle-5413-supported event output behaviors and statuses.

Table 7. PXIe-5413-Supported Events

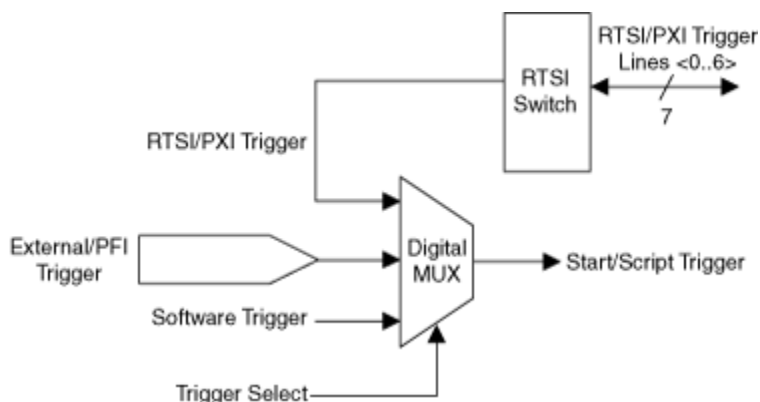
Event Name	Description	Output Behavior	Status
Ready for Start	Indicates that the signal generator is configured and ready to receive a start trigger.	Level	Live
Started	Indicates when the signal generator has received a start trigger and is generating a waveform.	Level, Pulse	Latched
Marker	A marker is an event that the device generates in relation to a waveform that is generated. The event is configured to occur at the time that a specific location or sample n if the waveform generates on the analog output connector. If the waveform loops multiple times in a segment, the marker generates each time the waveform loops.	Pulse, Toggle	Latched, Live
Done	Indicates that the generation of the previous waveform is complete.	Level, Pulse	Latched

Trigger Sources

The PXIe-5413 supports the following possible trigger sources:

- Immediate, also known as software
- Digital front panel connector
- PXI trigger bus lines (backplane of PXI devices)
- PXI star trigger line
- PFI inputs (SMA front panel connectors)

The following figure shows the trigger sources for the PXIe-5413.



For the PXle-5413, trigger sources are software selectable. To set a trigger source, use the parameters in the niFGEN Configure Trigger VI or the niFgen_ConfigureTriggerSource function.

Trigger Modes

The **frequency List**, **arbitrary waveform**, and **arbitrary sequence** output modes on the PXle-5413 support the following trigger modes that affect the behavior of the signal generator when receiving triggers.

- Single
- Continuous
- Stepped
- Burst

Trigger Types

You can trigger your signal generator with one of the following types of triggers:

- Digital Edge
- Digital Level
- Software Edge

Refer to the following table for the triggers supported by PXle-5413.

Table 8. PXle-5413-Supported Triggers

Trigger Name	Supported Types	Description
Start	• Digital Edge • Software Edge	The start trigger transitions a device from an idle state to a generation state where the device can respond to sample clocks.
Script	• Digital Edge	The script trigger is a general-purpose trigger with a role that is entirely determined by the context of the generation script. A script allows you to create sophisticated generation operations. For example, the script could configure the device to generate waveform A, then wait for the script

Trigger Name	Supported Types	Description
	- Digital Level - Software Edge	trigger, then generate waveform B. You can create multiple script triggers for use in your application. After a digital edge script trigger has been received, that trigger remains true for all subsequent instructions until the clear instruction is called or the trigger is reset after being used in the wait, repeat/end repeat, or if instructions.

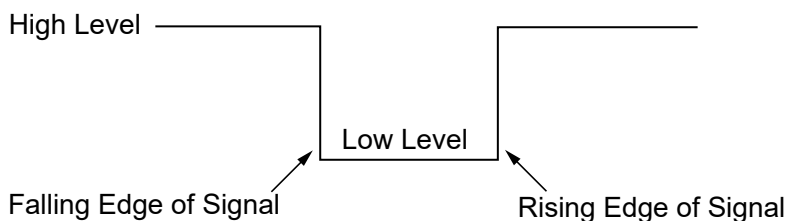
Digital Edge

Triggers configured to act on a rising or falling edge of a digital signal are called ***digital edge*** triggers.

A digital signal has two discrete levels: a high level and a low level. When the signal transitions from high to low or from low to high, a digital edge is created. There are two types of edges:

- Rising—Occurs when the signal transitions from low level to high level.
- Falling—Occurs with a transition from high level to low level

You can configure an edge trigger to occur either at the falling edge of signal or at rising edge of signal as the following figure shows.



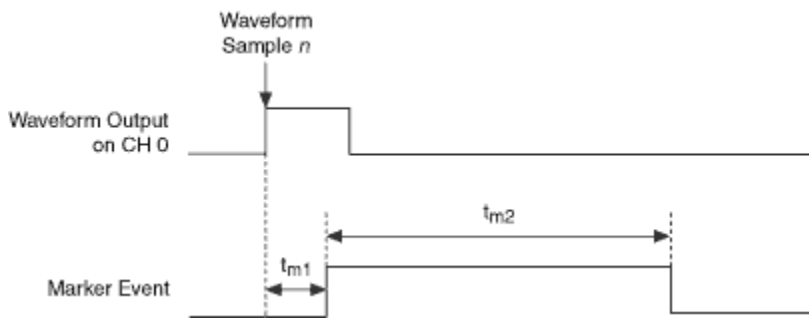
Digital Level

Triggers configured to act when a signal goes below the defined low level or above the defined high level are ***digital level*** triggers. Not all triggers can be configured to be level triggers.

Software Edge

A **software edge** trigger is generated internally by a programmatic call to the niFgen Send Software Edge Trigger VI or the niFgen_SendSoftwareEdgeTrigger function. Based on the conditions specified in the program, this trigger can occur at any time.

Marker Events



The previous figure shows a pulse that represents a waveform sample n that is one sample clock in width of a waveform being generated on the analog output connector. The second pulse, the marker event, represents the pulse that generates when the corresponding waveform sample n outputs at the analog output connector.

t_{m1} represents the delay in time of the marker event generated relative to the configured waveform sample n being generated.

t_{m2} represents the marker event pulse width in time.

NI-FGEN takes into account the factors that affect the delays in the digital and analog paths in assuring that the marker event appears within one sample clock of the waveform output. Therefore, t_{m1} is less than one sample clock period.

The marker event pulse width, t_{m2} , is at least 150 ns and can be significantly longer than 150 ns for slower sample clocks. You can configure the pulse width by setting the Marker Event Pulse Width property or the `NIFGEN_ATTR_MARKER_EVENT_PULSE_WIDTH` attribute. Instruments commonly have a minimum pulse width specification for a trigger to be registered, and trigger pulses of smaller width are ignored. The signal generator ensures that a minimum pulse width exists on the marker event by using a pulse stretching circuit. A sample clock rate of 100 MS/s has a period of 10 ns, requiring the pulse to be lengthened for

many devices to register the marker as a good trigger pulse.

Markers as Trigger Outputs

A delay of at least 44 sample clocks exists between the start trigger and the analog waveform generation on the output connector. Therefore, synchronizing the signal generator output signal to other devices with fast trigger response times is accomplished using the marker event from the signal generator as the trigger source for the other device for more precise alignment to the generating waveform. For the PXIe-5413, you can do this using the PXI trigger lines or PFI terminals.

Event Delays

Event delays can manually delay Marker, Started, and Done events so that they are aligned on a particular sample clock period.

Delay is applied to the event with respect to the analog output of the signal generator. For example, a delay of 0 sample clocks aligns the event with the analog output signal, while a delay of +2 sample clocks causes the event to appear two sample clock periods after the analog output appears. All event delays are adjusted in increments of sample clock periods, regardless of the units used to set the delay. For example, if you provide a value for delay with units of seconds, the delay is coerced up to the nearest sample clock period.

The event delay attributes must be set before waveform generation is initiated. Any changes made to other attributes during waveform generation may change the analog output delay. NI-FGEN does not compensate for this change in the analog output delay and continues to apply the event delay that you was originally configured.

If an event delay is applied to an event that is being exported to multiple output terminals, NI-FGEN aligns the event on the first terminal you specified.

You can configure event delays by calling the `Marker Event Delay` property or the `NIFGEN_ATTR_MARKER_EVENT_DELAY` attribute, the `Started Event Delay` property or the `NIFGEN_ATTR_STARTED_EVENT_DELAY` attribute, or the `Done Event Delay` property or the `NIFGEN_ATTR_DONE_EVENT_DELAY` attribute.

Synchronization

Synchronization occurs when two or more measurement devices operate in step with a common reference clock. The PXle-5413 supports synchronization using the NI-TClk API with a single NI-FGEN session per PXle-5413.



Note NI-TClk synchronization support for the PXle-5413 was first available in NI-FGEN 18.1.

You can synchronize the PXle-5413 with other instruments using PXle_Clk100 as the reference clock source and NI-TClk.

Use the NI-FGEN `Absolute Delay` property or `NIFGEN_ATTR_ABSOLUTE_DELAY` attribute to reduce trigger jitter or to maintain synchronization repeatability when synchronizing multiple instruments with NI-TClk. Phase DAC adjustment is not supported.

Synchronization with the Two-Channel PXle-5413

The channels of a two-channel PXle-5413 are automatically synchronized when they are in the same NI-FGEN session.

Synchronization with NI-TClk is not supported with multiple channel-based NI-FGEN sessions to a single two-channel PXle-5413.

Related concepts:

- [Reference Clock](#)

Streaming

Streaming is a way to generate waveforms that are too large to fit in the onboard memory of the signal generator. Streaming can be used in arbitrary waveform, arbitrary sequence, or script output modes.

To stream waveform data, allocate and identify all or a portion of the signal generator

onboard memory to act as an onboard waveform for streaming. Before initiating waveform generation, fill that onboard memory with the first part of your waveform. As the waveform is generated, continuously write the remaining waveform data until the waveform is complete.



Note Streaming should not be used if the data to be generated is less than or equal to the size of the streaming waveform.

Average Performance Rates

Average data transfer rates are highly system dependent. The following table is an example of the average sustainable transfer rates using 16-bit (or 2 byte) samples for PXle-5413.



Note Data transfer rates are limited by the speed of your streaming storage solution.

Table 9. PXle-5413 Average Performance Rate

Data Source	Data Rate (MB/s)	Note
From disk, such as a redundant array of independent disks (RAID).	> 600	The number is obtained using several file I/O optimizations.

Direct DMA

Direct DMA can be used to transfer waveform data to the signal generator onboard memory at rates well beyond the typical 5 to 30 MB/s range in a standard PC-based architecture. To achieve such high rates, Direct DMA establishes a direct connection between the signal generator onboard memory and a specialized waveform data source. Direct DMA is commonly used to stream waveform data from disk at data rates of greater than 100 MB/s.

In a standard PC-based architecture playback rates to the 5 to 40 MB/s range are limited by the following factors:

- Shared data paths through the PCI controller, I/O bus, CPU, and memory that

divide down bandwidth

- Latencies introduced by the operating system and application software managing data flow
- Nondeterministic operating system management of file I/O

For example, in a standard PC-based architecture, downloading a waveform to the onboard memory of your signal generator requires the following process for every 4,096 byte segment of physical memory:

1. The signal generator retrieves an address identifying the location of the data segment in the host memory.
2. The signal generator uses the address to request transfer of the data segment in the host memory.
3. The signal generator downloads the data to the onboard memory of the signal generator.
4. Steps 1-3 repeat every 4,096 bytes until the waveform data is fully downloaded.

To download a 16 MB waveform (16,777,216 bytes), this process repeats 4,096 times (16,777,216 bytes/4,096 bytes).

The direct connection established by Direct DMA to the data source is able to minimize or eliminate the factors that limit playback rates. To download data to the onboard memory of the signal generator with Direct DMA, the following process occurs only once:

1. The signal generator retrieves an address identifying the location of the waveform data source. This address generally refers to a large window of physically contiguous memory.
2. The signal generator uses the address to contact the waveform data source.
3. The signal generator requests the data segment and size from waveform data source. In this case, the size requested by the signal generator can be fairly large (for example, 16 MB).
4. The signal generator downloads the data to the onboard memory of the signal generator from the waveform data source in one operation.

The transfer process is much faster and more efficient because an address to the waveform data is requested once (in this case, the address is to the waveform data source).

PXI Express Bandwidth Considerations

NI PXI Express signal generators use PCI Express as the interface to the computer. The physical connection between a PXI Express signal generator and a computer is called a PCI Express link. When a signal generator generates a waveform, it can saturate this link. Saturation occurs when the signal generator is copying data to its onboard memory from computer memory as rapidly as possible, utilizing all of the bandwidth of the PCI Express link.

The theoretical bandwidth of the onboard memory of the signal generator is 3.2 GB/s, and the theoretical bandwidth of the PCIe link is approximately 850 MB/s. This difference in bandwidths means that transferring data to the onboard memory can saturate the PCI Express link. The waveform data download rate is dependent on the system configuration, with a maximum of approximately 600 MB/s.

High-speed transfers that saturate the PCI Express link can affect other instruments when multiple instruments share a single PCI Express link. The bandwidth sharing in this case becomes a problem if other instruments in the application require guaranteed bandwidth to the PCI Express bus.

You can configure the NI PXI Express signal generator to limit its use of the PCI Express bus, leaving bandwidth for other instruments that are more susceptible to short-term bursts of traffic on a shared PCI Express link. You can also improve the performance of your instrument based on your system configuration. The following sections list the properties and attributes that allow you to modify the way that the signal generator uses the PXI Express bus.

Maximum Bandwidth

Allows you to optimize bus bandwidth usage for multi-device streaming applications.

LabVIEW Property	C/C++ Attribute
Maximum Bandwidth	NIFGEN_ATTR_DATA_TRANSFER_MAXIMUM_BANDWIDTH

Maximum In-Flight Read Requests

Allows you to specify, based on the requirements of your application, the maximum

number of in-flight read requests for data transfers.

LabVIEW Property	C/C++ Attribute
Maximum In-Flight Read Requests	NIFGEN_ATTR_DATA_TRANSFER_MAXIMUM_IN_FLIGHT_READ_REQUESTS

Preferred Packet Size

Allows you to configure, based on the requirements of your system, the preferred size of the data field in the PCI Express data transfer packet.

LabVIEW Property	C/C++ Attribute
Preferred Packet Size	NIFGEN_ATTR_DATA_TRANSFER_PREFERRED_PACKET_SIZE

Frequency Domain Fundamentals

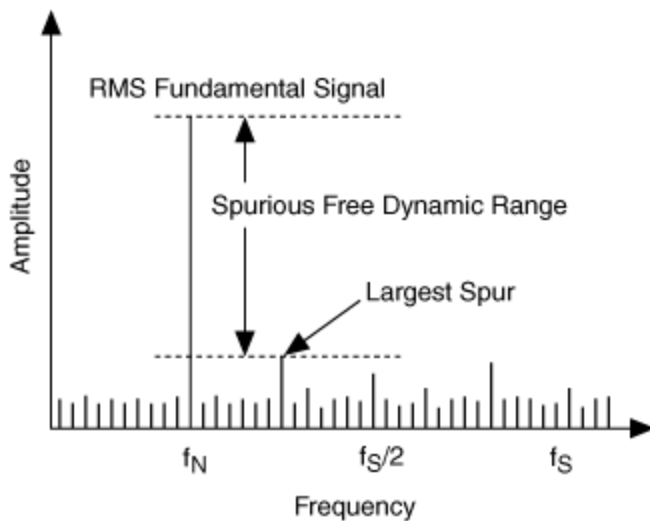
Refer to the following topics to understand the PXIe-5413 frequency domain.

SFDR

Spurious-free dynamic range (SFDR) is the usable dynamic range before spurious noise interferes with or distorts the fundamental signal.

For specification purposes, the amplitude of the fundamental signal is usually -1 dBFS. SFDR is the measure of the difference in amplitude between the fundamental signal and the largest harmonically or nonharmonically related spur from DC to the full Nyquist bandwidth (half the sampling rate). A spur is any frequency bin on a spectrum analyzer, or from a Fourier transform, of the analog signal. SFDR is expressed in dBc.

The following figure illustrates how SFDR is measured.



THD

The total harmonic distortion (THD) of a signal is the ratio of the sum of the powers of the first five harmonics above the measured fundamental frequency to the power of the fundamental frequency. THD is usually expressed in dB or dBc. Measurements for calculating the THD are made at the output of a device under specified conditions.

SINAD

Signal-to-noise-and-distortion ratio (SINAD) is the ratio of the RMS signal amplitude to the RMS sum of all other spectral components, including the harmonics but excluding DC. SINAD is usually expressed in dB.

ENOB

Effective number of bits (ENOB) is another way of specifying signal-to-noise and distortion ratio (**SINAD**). ENOB is calculated using the following formula:

$$\text{ENOB} = \frac{\text{SINAD} - 1.76}{6.02}$$



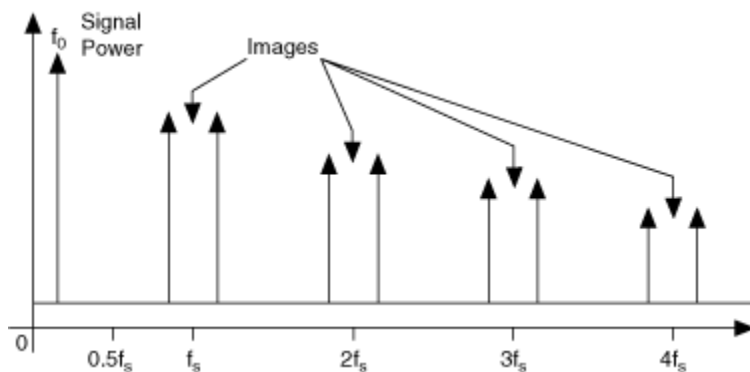
Note This equation assumes the full scale of the DAC is utilized.

The ENOB value is the value of an ideal DAC that is equivalent to the DAC of the device.

Filtering and Interpolation

All NI signal generators use a digital-to-analog converter (DAC) to generate the signals or waveforms that eventually appear at the output connector. These generated signals have a number of discrete voltage levels dependent on the number of bits in the DAC.

A digital waveform must be updated at least twice as fast as the bandwidth of the desired analog signal to be accurately generated (Shannon's Sampling Theorem). Even though the theoretical requirement for sample clock, f_s , is twice that of the signal bandwidth, f_0 , images are introduced in the output signal at $|f_0 \pm nf_s|$, as shown in the following figure.



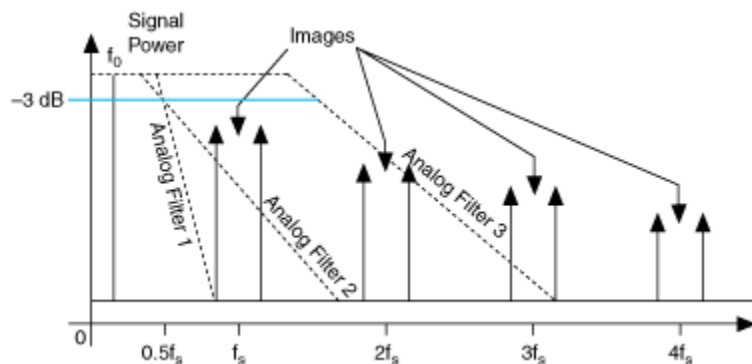
The images in the previous figure degrade the spectral purity of the signal, creating the need to filter these images out of the signal.

To create quality signals, all NI signal generators can lowpass filter the generated signal. A lowpass filter can smooth the raw DAC output. The filter removes high-frequency aliased components that are introduced through the digital generation of the signal. You can implement the lowpass filter through both analog and digital filters.

Designing an analog filter that rejects the images and yet gets maximum output bandwidth (0 to $0.43f_s$) is difficult and is represented by the curve Analog Filter 1 in the following figure. Analog Filter 2 represents a more practical filter. This filter is not as aggressive as Analog Filter 1. Analog Filter 2 does not filter out the images near f_s , but it does reject all the others. Analog filters have trade-offs between the roll-off of the attenuation after the 3 dB point and the flatness of the attenuation before the 3 dB point.

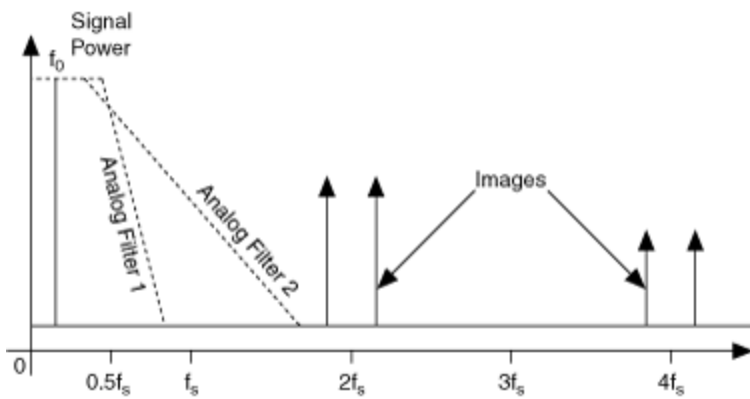
Another aspect of the analog filter is group delay—the amount of time needed for a signal having finite time duration, such as a pulse, to pass through the analog filter. Ideally, in an analog filter with linear group delay, all frequencies present in the signal should have the same time delay so that the signal is not distorted.

The third filter, Analog Filter 3, has a much higher 3 dB point than the first two analog filters. Because of the higher 3 dB point, the filter is very nearly flat in the passband (0 to $0.43f_s$). Analog Filter 3 does not filter the images produced at f_s and $2f_s$ at all, but this shortcoming can be alleviated with a digital interpolation filter.

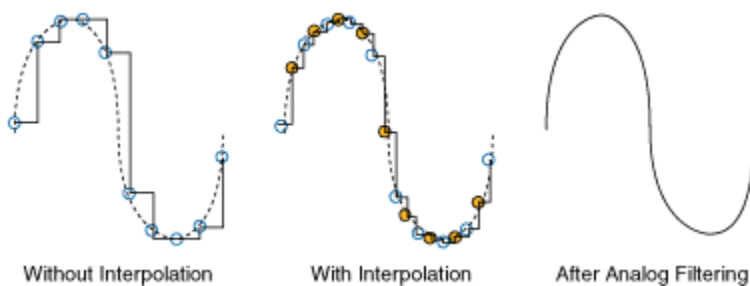


To ease the requirements of the analog filter and to get more output bandwidth, NI signal generators use a halfband digital filter to interpolate one, three, or seven samples between every two waveform samples at two times, four times, and eight times the sample frequency, f_s . Also, the DAC operates at an effective sampling rate that is two times ($2f_s$), four times ($4f_s$), and eight times ($8f_s$) the sample frequency—specifically, the rate at which the data is clocked from the memory into the DAC.

In the following figure, the two times interpolating filter is used and the effective sample rate of the DAC is $2f_s$. The images at $f_s \pm f_0$ are no longer an issue, and the images are now at $|2f_s \pm f_0|$.



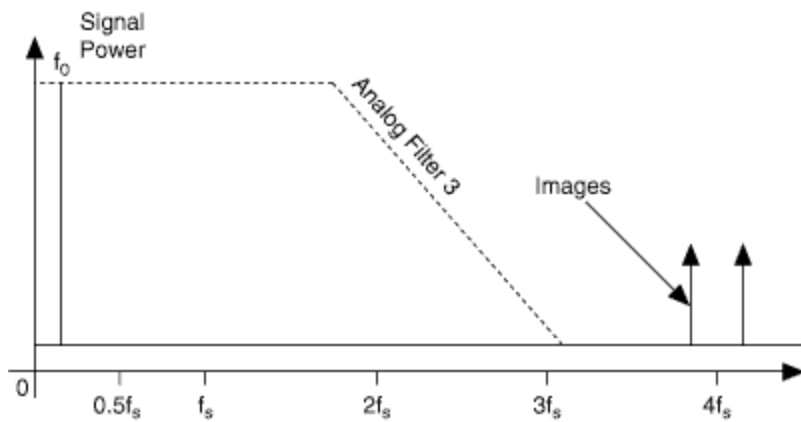
Now, Analog Filter 2 can easily filter out all the images due to the digital generation of the signal. This behavior is seen in the frequency domain representation in the previous figure and in the time domain representation in the following figure.



Note The allowable range of interpolation factors is dependent on the NI signal generator being used.

Using two times interpolation filtering with a DAC effective sample rate of $2f_s$ eliminates images well and generates a good signal. However, increasing the interpolation filter to 4 further improves the output signal.

The following figure shows a signal image with four times interpolation, and the effective DAC sample rate at $4f_s$. The images are shifted up to $4f_s$, and well above the cutoff frequency of Analog Filter 3. This configuration eliminates the spectral images and has a filter that is maximally flat within the passband. This configuration approaches an ideal design in digitally generating spectrally pure waveforms.



To generate the most spectrally pure signals using the digital filter, use the highest interpolation factor possible.

Power On and Reset

The PXIe-5413 has different settings when powering on and after powered on or a reset.

The PXIe-5413 is in the following state from the time at which the computer begins to power on until the operating system has started and NI-FGEN is loaded.

Table 10. PXIe-5413 State During Power-On

CH <0..1>	PFI <0..1>	PXI_Trig <0..7>
- Analog output is disabled. 0 V amplitude 50 Ω output impedance to ground	Tristated	Tristated and floating

When the operating system has started and NI-FGEN is loaded or after performing a device reset, the PXIe-5413 is in the following state.

Table 11. PXIe-5413 State After Power-On or Reset

CH <0..1>	PFI <0..1>	PXI_Trig <0..7>	Clocks
- Analog output is enabled. 0 V amplitude 0 dB attenuation - Low-gain amplifier path is enabled.	Tristated	Tristated and floating	Sample Clock Timebase is set to use the internal voltage-controlled crystal oscillator (VCXO).

Integration and System Considerations

Refer to the following topics for information about integrating the PXIe-5413 into a PXI-based measurement system.

Environment

Device performance and reliability may be limited at temperatures above the specified operating range. For best performance take the following precautions:

- Ensure that the ambient temperature is within the specifications for the device and is stable (± 5 °C).
- Follow standard metrology practices.
- Use a PXI or PXI Express chassis with a well-designed cooling system.

Operating NI PXI and PXI Express signal generators outside the specified operating temperatures can increase bias currents in the electronic components, increase noise, accelerate drifts, and decrease product life. Beyond the maximum specified operating temperatures, the device may perform differently than during factory calibration, resulting in additional measurement errors. Also, operating the device outside of the humidity specification (>80%, >35 °C) may cause leakages between circuit components and introduce measurement error.

To optimize cooling and ensure best performance and reliability the use the following guidelines:

- Chassis that provide multiple fan speed settings should always be run with fans set on high or auto if applicable to your chassis. Never set the fans to low or turn them off.



Note In newer NI chassis the settings are HIGH and AUTO, in some older NI chassis the fan settings may be HI and LO.

- All empty slots in the chassis should be covered with a blank slot filler panel.
- Remove and clean the inlet filters often to prevent buildup of dust and other foreign material that may restrict airflow.
- The chassis should be located such that the fan inlets and outlet vents are not obstructed. Other objects and equipment should be kept a minimum of 3 inches from the fan inlets.

For more information about forced-air cooling, refer to your chassis documentation.

For more information about environment requirements, refer to **PXIe-5413 Specifications**.

PXI/PXI Express Chassis Cooling

Not all PXI or PXI Express chassis provide the same cooling. When selecting a PXI or PXI Express chassis, consideration should be given to providing adequate airflow for high power and sensitive devices such as NI signal generators.

NI PXI and PXI Express signal generators are high-precision instruments and may be sensitive to interference from other electronic devices. To optimize the accuracy and performance of the device, you may need to locate the device in a slot away from devices with power supplies and other noisy circuitry. The device may also be sensitive to heat generated by high-power products in neighboring slots. When possible, consider locating the device away from high-power devices to optimize cooling.

Using the PXIe-5413 with CompactPCI Express Products

The CompactPCI/PXI Express backplane integrates PCI Express while still preserving compatibility with current PXI modules. PXI Express hybrid slots are capable of delivering signals for both PCI and PCI Express. Thus, the hybrid slot allows you to

install a PXI module that uses PCI signaling or a PXI Express module that uses PCI Express signaling.

PXI Star Trigger Lines

The PXIe-5413 can receive a signal from the PXI star if placed in a peripheral slot.

Additionally, an NI in a peripheral slot can be configured to route a signal onto the PXI star line if the PXI star trigger controller is configured to receive it.



Note The PXIe-5413 in the PXI Express chassis can import clocks on the PXI star line, but cannot import triggers, or export clocks or triggers over the PXI star line.

System Reference Clock, PXIe_CLK100

The typical PXIe chassis supplies PXI_CLK10, PXIe_CLK100, and PXIe_SYNC100 to every peripheral slot with an independent low skew driver for each signal. As a PXIe peripheral module, the PXIe-5413 uses PXIe_CLK100 and PXIe_SYNC100 for clocking synchronization.

PFI Lines

PFI lines are multipurpose programmable function input/outputs. These lines serve as connections to virtually all internal timing signals. The PFI lines of NI signal generators can accept or generate a trigger, generate a marker, accept or generate a reference clock. The function of each PFI line is independent.

If you are using LabVIEW to program your signal generator and you want to connect external signal generators to the PFI lines, you can use the niFgen Configure Sample Clock Source VI, the niFgen Configure Reference Clock VI, or the niFgen Configure Trigger VI to route external signals to internal sources. You can use the niFgen Export Signal VI to route internal signals to the PFI lines on the front panel.

If you are using LabWindows/CVI to program your signal generator, you can use the niFgen_ConfigureSampleClockSource function, the niFgen_ConfigureReferenceClock function, or the niFgen_ConfigureTriggerSource function to route external signals to internal sources. You can use the niFgen_ExportSignal function to route internal

signals to the PFI lines on the front panel.



Notice If you enable a PFI line for output, do not connect any external signal source to it; doing so can damage the device, the computer, and the connected equipment.

MXI Optimization Application

MXI-3

If you are using the MXI-3 interface to control the PXI chassis, the MXI-3 Optimization Application must be run prior to using the NI signal generator. By default, this application runs automatically when Windows starts. If you have an initialization, timeout, or performance issue with your module, or if you are not certain that the application ran, select **Start»All Programs»National Instruments MXI-3»MXI-3 Optimization** to run the application. If you continue to have initialization or performance issues, refer to the MXI-3 documentation at **Start»All Programs»National Instruments MXI-3**, or visit NI Technical Support at ni.com/support.

MXI-4 and MXI-Express Optimization

Optimization for MXI-4 and MXI Express are performed automatically by the hardware.

Finding Examples

NI-FGEN includes several examples that demonstrate basic signal generator applications. Access examples depending on your programming environment and the driver APIs you use to control your instrument.

NI Example Finder

The NI Example Finder is a utility that organizes examples into categories and allows you to browse and search installed examples. For example, search for "FGEN" to locate all waveform generator examples. You can see descriptions and compatible hardware models for each example or see all the examples compatible with one particular hardware model.

To locate examples using the NI Example Finder within LabVIEW, select **Help » Find Examples** and navigate to **Hardware Input and Output » Modular Instruments**.

Installed Example Locations

The installation location for NI-FGEN example programs differs by programming language and development environment. You can access the examples through the Start menu, by navigating to **Start » All Programs » National Instruments » » Examples**.

Refer to the following table for information about example program installation locations.

Table 12. Installed NI-FGEN Example Locations

Option	Installed Example Location
LabVIEW	<LabVIEW>\examples\instr\niFgen, where <LabVIEW> is the directory for the specific LabVIEW version that is installed.
Visual C/C++	Users\Public\Documents\National Instruments\NI-FGEN\NI-FGEN Examples\c
LabWindows/CVI	Users\Public\Documents\National Instruments\CVI\Samples\IVI\IviFgen

Option		Installed Example Location
.NET	4.0	Users\Public\Documents\National Instruments\NI-FGEN\NI-FGEN Examples\DotNET 4.0
	4.5	Users\Public\Documents\National Instruments\NI-FGEN\NI-FGEN Examples\DotNET 4.5
Python		Refer to NI-FGEN Python documentation for details.

Refer to the ***NI-FGEN Tutorial*** section in ***NI-FGEN User Manual*** to understand the basic programming steps.

Common Example Programs

NI-FGEN includes a collection of programming examples to help you get started developing an application. You can modify example code and save it in an application. You can use examples to develop a new application or add example code to an existing application.

The following NI-FGEN example programs supported on the PXIe-5413 demonstrate common functions and operations:

- **Fgen Basic Arb Waveform**—Demonstrates how to configure your instrument to generate a basic arbitrary waveform.
- **Fgen Basic Standard Function**—Demonstrates how to configure your instrument to generate a basic standard waveform.

PXle-5413 Installation

Complete the following steps to install the PXle-5413 into a chassis and prepare it for use.

1. [Unpacking the Kit](#)

Take precautions to prevent electrostatic discharge when unpacking and inspecting your hardware.

2. [Installing the Software](#)

3. [Installing the PXle-5413 into a Chassis](#)

4. [Performing Initial Hardware Configuration](#)

5. [Self-Calibrating the PXle-5413](#)

Self-calibration adjusts the PXle-5413 for variations in the module environment. Perform a complete self-calibration after the first time you install the PXle-5413.

Unpacking the Kit

Take precautions to prevent electrostatic discharge when unpacking and inspecting your hardware.



Notice To prevent electrostatic discharge (ESD) from damaging the device, ground yourself using a grounding strap or by holding a grounded object, such as your computer chassis.

1. Touch the antistatic package to a metal part of the computer chassis.
2. Remove the device from the package and inspect the device for loose components or any other sign of damage.



Notice Never touch the exposed pins of connectors.



Note Do not install a device if it appears damaged in any way.

3. Unpack any other items and documentation from the kit.

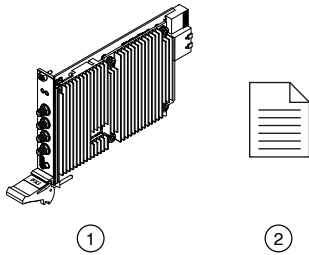


Note Store the device in the antistatic package when the device is not in use.

Kit Contents

Refer to the following figure to identify the contents of the PXIe-5413 kit.

Figure 11. PXIe-5413 Kit Contents



1. PXIe-5413 Module
2. Documentation

Installing the Software

You must be an Administrator to install NI software on your computer.

1. Install an ADE, such as LabVIEW or LabWindows™/CVI™.
2. Download the driver software installer from ni.com/downloads.
Package Manager downloads with the driver software to handle the installation. Refer to the Package Manager Manual for more information about installing, removing, and upgrading NI software using Package Manager.
3. Follow the instructions in the installation prompts.



Note Windows users may see access and security messages during installation. Accept the prompts to complete the installation.

4. When the installer completes, select **Restart** in the dialog box that prompts you to restart, shut down, or restart later.

PXIe-5413 Firmware Licensing

Software is licensed pursuant to the software license agreements provided with the

software or, in the absence of such license agreements, the NI Software License Agreement available at ni.com/legal at the time of purchase.

For a copy of the software licenses applicable to the firmware for this hardware product, as well as copies of third party licenses and any required notices shipped with the firmware, visit ni.com/info and enter the Info Code .

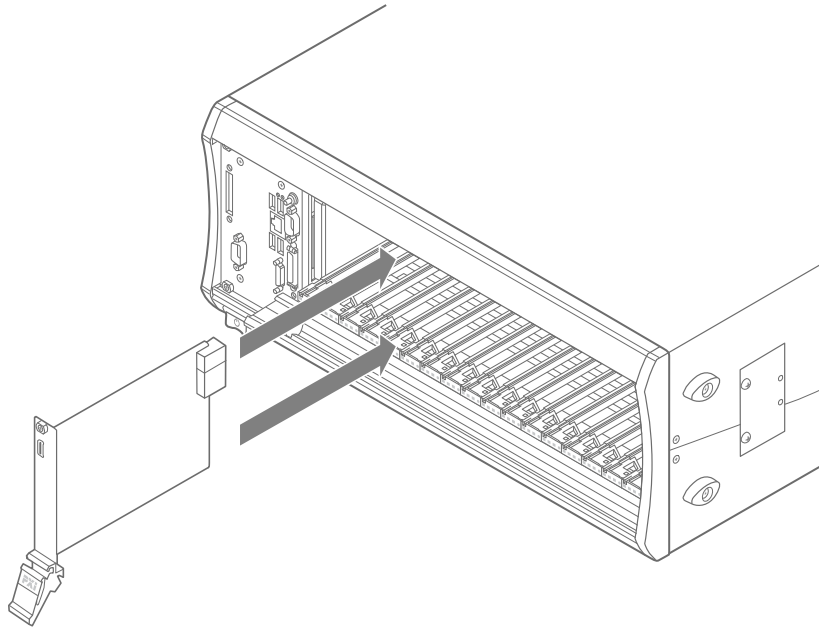
Installing the PXIe-5413 into a Chassis



Notice To avoid damage from electrostatic discharge (ESD) or contamination, handle the module only by its edges or the metal bracket.

1. Ensure that the AC power source is connected to the chassis before installing the module.
The AC power cord grounds the chassis and protects it from electrical damage while you install the module.
2. Power off the chassis.
3. Inspect the slot pins on the chassis backplane for any bends or damage prior to installation. Do not install a module that has a damaged backplane.
4. Position the chassis to ensure that nothing obstructs the inlet vents and outlet vents.
For more information about optimal chassis positioning, refer to the chassis documentation.
5. Remove the black plastic covers from all the captive screws on the module front panel.
6. Identify a supported slot in the chassis. You can place the PXIe-5413 module in the following slot types.
 - PXI Express hybrid peripheral slots (●^H).
 - PXI Express system timing slots (◻●).
 - PXI Express peripheral slots (●).
7. Touch any metal part of the chassis to discharge static electricity.
8. Ensure that the ejector handle is in the downward (unlatched) position.

Figure 12. Module Installation



9. Place the module edges into the module guides at the top and bottom of the chassis. Slide the module into the slot in full.
10. Latch the module in place by pulling up on the ejector handle.
11. Secure the module front panel to the chassis using the front-panel mounting screws.



Note Attach all front panel screws to the PXI chassis to ensure that all backplane connectors are fully mated. Tightened top and bottom mounting screws increase mechanical stability. Increased mechanical stability also electrically connects the front panel to the chassis that can improve the signal quality and electromagnetic performance.

12. Cover all empty slots using filler panels (standard or EMC). Alternatively, use slot blockers with filler panels.



Note For more information about installing slot blockers and filler panels, go to slot blocker documentation.

Performing Initial Hardware Configuration

Before using the PXIe-5413, use Hardware Configuration Utility or Measurement & Automation Explorer (MAX) to perform initial hardware configuration.

Configuring the PXle-5413 in Hardware Configuration Utility

NI recommends using Hardware Configuration Utility to perform initial hardware configuration.

1. Open Hardware Configuration Utility.
The PXle-5413 should appear in the system pane automatically.
2. Record the name Hardware Configuration Utility assigns to the PXle-5413 or, if desired, provide a custom name to the PXle-5413.
Use this name when programming the PXle-5413.
3. Validate that your instrument is installed correctly:
 - a. Select the PXle-5413 module in the system pane and expand the **Troubleshooting** area of the configuration pane.
 - b. Click **Self-test**.
 Hardware Configuration Utility reports when the hardware setup is validated.

Configuring the PXle-5413 in MAX

Use Measurement & Automation Explorer (MAX) to configure your NI hardware. MAX informs other programs about which NI hardware products are in the system and how they are configured. MAX is automatically installed with NI-FGEN.

1. Launch MAX.
2. In the configuration tree, expand **Devices and Interfaces** to see the list of installed NI hardware.
Installed modules appear under the name of their associated chassis.
3. Expand your **Chassis** tree item.
MAX lists all modules installed in the chassis. Your default names may vary.



Note If you do not see your module listed, press <F5> to refresh the list of installed modules. If the module is still not listed, power off the system, ensure the module is correctly installed, and restart.

4. Record the identifier MAX assigns to the hardware. Use this identifier when programming the PXle-5413.
5. Self-test the hardware by selecting the item in the configuration tree and clicking **Self-Test** in the MAX toolbar.
MAX self-test performs a basic verification of hardware resources.

My Hardware Doesn't Appear in Hardware Configuration Utility or MAX

1. Check that the ACCESS/ACC LED on the hardware is solid green.
2. Check if the connection between the hardware and software is working.

Software	Description
Hardware Configuration Utility	Click the refresh button ().
MAX	a. In the MAX configuration tree, expand Devices and Interfaces. b. Expand the Chassis tree to see the list of installed hardware. Then press F5 to refresh the list.

3. If the instrument is still not listed, power off the system. Ensure that all hardware is correctly installed, and restart the system.
4. Navigate to the Device Manager. Right-click the **Start** button and select **Device Manager**.
5. Depending on your controller type, verify Device Manager settings.

Controller Type	Description
PXI controller	a. Verify that a National Instruments entry appears in the system device list. b. If error conditions appear in the list, right-click the module you are using in the Device Manager and select Update Driver.
MXI controller	Right-click PCI-to-PCI Bridge and select Properties from the shortcut menu to verify the bridge status.

6. Restart your computer.

What Should I Do if the PXIe-5413 Fails the Self-Test?

1. Call the niFgen Reset Device VI or niFgen_ResetDevice function to reset the

PXle-5413 and then perform the self-test again.

You can also perform a self-test in Hardware Configuration Utility or MAX.

2. Restart the system, and then perform the self-test again.
3. Power off the chassis.
4. Reinstall the failed module in a different slot.
5. Power on the chassis.
6. Perform the self-test again.

Self-Calibrating the PXle-5413

Self-calibration adjusts the PXle-5413 for variations in the module environment. Perform a complete self-calibration after the first time you install the PXle-5413.

The PXle-5413 modules are externally calibrated at the factory but you should perform a self-calibration in all of the following situations:

- After first installing the PXle-5413 in a chassis
 - After any module that is in the same chassis as the PXle-5413 is installed, uninstalled, or moved
 - When the PXle-5413 is in an environment where the ambient temperature changes. Refer to the module specifications to find the allowable ambient temperature for your instrument.
 - When the PXle-5413 temperature has drifted from T_{cal} by more than specified since the last self-calibration. Refer to the module specifications to find the allowable difference from T_{cal} for this instrument.
 - More than 24 hours have passed since the previous self-calibration
1. Install the PXle-5413 and let it warm up for the recommended warm-up time listed in the module specifications.



Note Warm up begins when the chassis has been powered on and the operating system has completely loaded.

2. Self-calibrate the PXle-5413 by clicking the **Self-Calibrate** button in Hardware Configuration Utility or MAX, or calling niFgen Self Cal or niFgen_SelfCal.

Signal Connections to the PXIe-5413

The PXIe-5413 can connect to digital signals.

SCB-19 Pinout

NI recommends using the SCB-19 connector block to connect digital signals to the AUX 0 connector on the PXIe-5413 front panel.

Figure 13. SCB-19 Pinout

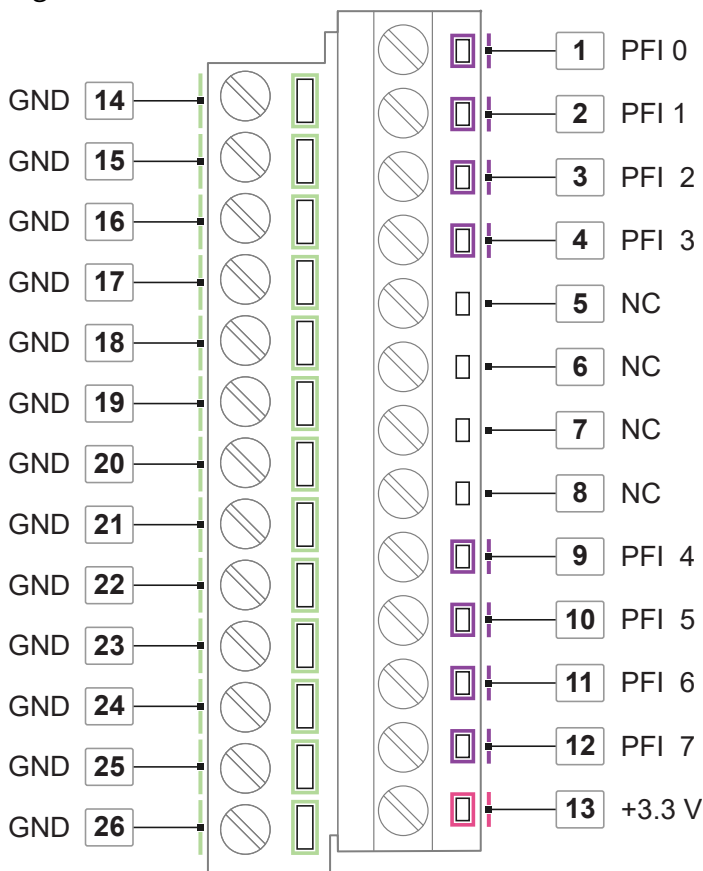


Table 13. SCB-19 Signal Descriptions

Signal Name	Description
PFI <0..7>	Bidirectional PFI line
NC	No connection
+3.3 V	+3.3 V power (200 mA maximum)

Signal Name	Description
GND	Ground reference for signals

PXIe-5413 Routes

You can create routes in the following tables with the PXIe-5413.

Table 14. PXIe-5413 Device Routes

Source	Destination			
	PXI_Trig Terminals	PFI Terminals	Triggers	Sync Pulse
PXI_Trig Terminals	N/A	N/A	✓	✓
PFI Terminals	N/A	N/A	✓	N/A
Events	✓	✓	✓	N/A
Triggers	✓	✓	✓	N/A
Sync Pulse	✓	N/A	N/A	N/A

Table 15. Source and Destination Terminal Values

Source and Destination	Terminal Values
PXI_Trig Terminals	PXI_Trig0 PXI_Trig1 PXI_Trig2 PXI_Trig3 PXI_Trig4 PXI_Trig5 PXI_Trig6 PXI_Trig7
PFI Terminals	PFI0 PFI1 AUX0/PFI0

Source and Destination	Terminal Values
	AUX0/PFI1 AUX0/PFI2 AUX0/PFI3 AUX0/PFI4 AUX0/PFI5 AUX0/PFI6 AUX0/PFI7
Events	0/ReadyForStartEvent 0/StartedEvent 0/DoneEvent 0/Marker0Event 0/Marker1Event 0/Marker2Event 0/Marker3Event 1/ReadyForStartEvent 1/StartedEvent 1/DoneEvent 1/Marker0Event 1/Marker1Event 1/Marker2Event 1/Marker3Event
Triggers	0/StartTrigger 0/ScriptTrigger0 0/ScriptTrigger1 0/ScriptTrigger2 0/ScriptTrigger3 1/StartTrigger 1/ScriptTrigger0 1/ScriptTrigger1 1/ScriptTrigger2 1/ScriptTrigger3
Sync Pulse	Clock10Pulse



Note Values beginning with 1 / are supported by only the PXIe-5413 (two-channel).

PXIe-5413 Configuration

Refer to the following topics to understand how to configure your instrument for waveform generation based on your needs.

Configuring an Internal Sample Clock

You can use the internal (onboard) sample clock to control the clocking rates of your signal generator. In arbitrary waveform or arbitrary sequence mode, you can choose a clock mode that allows you to configure the rate at which this sample clock runs.



Note The onboard clock is the default sample clock source. If you have not changed the settings for your sample clock source, you do not need to perform this task.

1. Call the niFgen Configure Sample Clock Source VI with `Source` set to `Onboard Clock` or call the `niFgen_ConfigureSampleClockSource` function with the `sampleClockSource` parameter set to `OnboardClock`.
2. If your application is using arbitrary waveform or arbitrary sequence output mode, call the niFgen Configure Clock Mode VI with `Clock Mode` set to the clock mode required for your application or call the `niFgen_ConfigureClockMode` function with the `clockMode` parameter set to the clock mode required for your application.

Configuring a Reference Clock

In a phase-locked loop, specify the source for the reference clock so that the frequency stability and accuracy of the sample clock timebase matches that of the reference clock.

Call the niFgen Configure Reference Clock VI or the `niFgen_configureReferenceClock` function and set the `Source` or `referenceClockSource` parameter respectively to the reference clock source. For example, set the `Source` or `referenceClockSource` parameter to `ClkIn` to obtain the reference clock signal from the Clk In front panel connector; set `Reference Clock Frequency` or

`referenceClockFrequency` to the frequency of the reference clock.

Configuring Triggers

1. Call the niFgen Configure Trigger Mode VI or the `niFgen_configureTriggerMode` function to select the trigger mode in which your trigger will generate. This mode affects the behavior of the trigger and depends on the output mode of the signal generator.
2. Call the niFgen Configure Trigger poly VI and choose the instance that corresponds to the type of trigger you want to use, or call one of the following niFgen Configure Trigger functions:
 - `niFgen_ConfigureDigitalEdgeStartTrigger`
 - `niFgen_ConfigureSoftwareEdgeStartTrigger`
 - `niFgen_ConfigureDigitalEdgeScriptTrigger`
 - `niFgen_ConfigureDigitalLevelScriptTrigger`
 - `niFgen_ConfigureSoftwareEdgeScriptTrigger`

Related reference:

- [Triggers and Events](#)

Creating a Marker Event

You can specify a marker and its location by setting an offset location value (in number of samples) from the start of the waveform. If the offset is out of range of the number of samples in that segment, PXle-5413 returns an error.

There are two rules for marker placement:

1. A marker can be specified only at offsets that are multiples of four samples (or two complex samples).
2. A marker must be placed at least four samples from the end of the waveform. In Burst trigger mode, a marker must be placed at least eight samples from the end of the waveform.

For example, for a waveform containing 100 samples, a marker at an offset of 0 or 4 is valid, but a marker at an offset of 3 is invalid. In addition, a marker at an offset of 97 or

100 is always invalid, while a marker at an offset of 96 is valid for all trigger modes except Burst. The marker can be placed at an offset of 92 for all trigger modes.

Related reference:

- [Triggers and Events](#)

Creating a Marker Event in Arbitrary Waveform Mode



Note NI-FGEN supports only Marker0 in arbitrary waveform mode.

1. Specify the active channel as `Marker0`.
2. Specify the marker event position by setting the `Arbitrary Waveform Marker Position` property or the `NIFGEN_ATTR_ARB_MARKER_POSITION` attribute.
3. Export the marker event by calling the `niFgen Export Signal VI` or the `niFgen_ExportSignal` function and setting the `Signal` or `signal` parameter respectively to `NIFGEN_VAL_MARKER_EVENT`.

Creating a Marker Event in Arbitrary Sequence Mode

1. Call the `niFgen Create Advanced Arb Sequence VI` or the `niFgen_CreateAdvancedArbSequence` function and set the `Marker Location Array` or `markerLocationArray` parameter respectively to the location at which you would like the marker to generate.
2. Export the marker event by calling the `niFgen Export Signal VI` or the `niFgen_ExportSignal` function and setting the `Signal` or `signal` parameter respectively to `NIFGEN_VAL_MARKER_EVENT`.

Creating a Marker Event in Script Mode

In script mode, you can create up to four markers for each waveform. For examples of scripting applications, refer to common scripting use cases in ***NI-FGEN User Manual***.

1. Export the marker event by calling the `niFgen Export Signal VI` or the `niFgen_ExportSignal` function.

2. Set the `Signal` or `signal` parameter respectively to `NIFGEN_VAL_MARKER_EVENT` and set the `Signal Identifier` or `signalIdentifier` parameter respectively to the name of the marker to export.

Writing a Script

Complete the following steps to write a script:

1. Call the `niFgen Configure Output Mode VI` or the `niFgen_ConfigureOutputMode` function to switch to script mode.
2. Write all waveforms that are referenced in the script by calling the `niFgen Write Named Waveform VI` or one of the `niFgen Write Named Waveform` functions and associate the proper names to them.
3. After your waveforms are written to your instrument, call the `niFgen Write Script VI` or the `niFgen_WriteScript` function to write the scripts containing the generation instructions to execute.

The script you write can manage waveform generation based on multiple waveforms and triggers. The following code example instructs to download waveforms A, B, C, and D into instrument memory, and then write a script that waits for a trigger to initiate generation and, when receiving this trigger, generates waveform A three times with a marker at position 16 each time and finally generates waveforms B, C, and D twice (BCDBCD).

```
script myFirstScript
  wait until scriptTrigger0
  repeat 3
    generate waveformA marker0(16)
  end repeat
  repeat 2
    generate waveformB
    generate waveformC
    generate waveformD
  end repeat
end script
```

4. You can write multiple scripts that exist simultaneously on your instrument. If you write multiple scripts to your instrument, you must select the one you want to execute by setting the `Script to Generate` property or the `NIFGEN_ATTR_SCRIPT_TO_GENERATE` attribute.

5. Call the niFgen Initiate Generation VI or the niFgen_InitiateGeneration function to execute the selected script.



Note Internally, the script stores physical instrument memory locations to refer to the named waveforms. Thus, you must write all waveforms to the instrument before writing the script, or the instrument does not know where the waveform is located. The niFgen Initiate Generation VI or the niFgen_InitiateGeneration function returns an error if this rule is violated. If you delete waveforms and rewrite them, rewrite the script to update it with the new locations even if the script text has not changed.

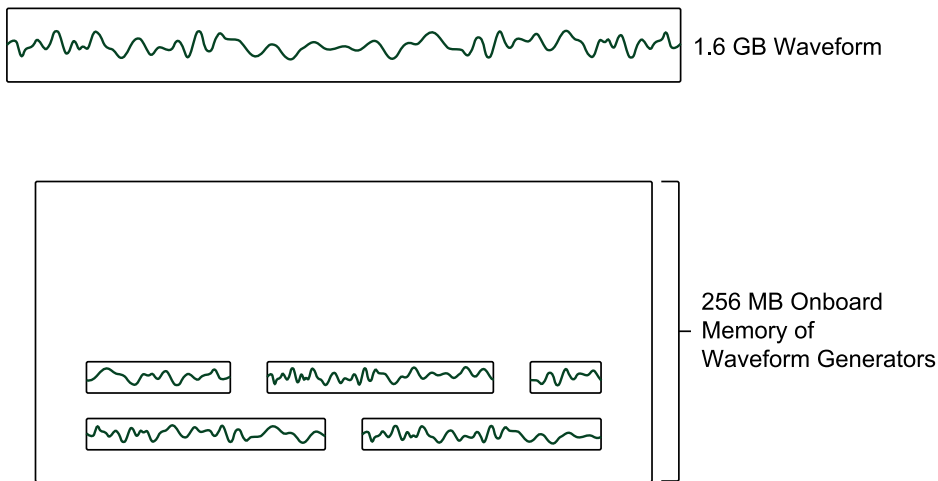
In some cases at high sample clock rates, your script may result in an underflow error. Underflow errors in a script can be created by waveform size, marker placement, and specific script instructions. To avoid underflow errors, refer to the minimum waveform size in **PXle-5413 Specifications**.

Streaming Waveform Data

NI-FGEN provides the following programmatic examples:

- LabVIEW: `Fgen Arb Waveform Streaming.vi`
- LabWindows/CVI: `ArbitraryWaveformStreaming.prj`

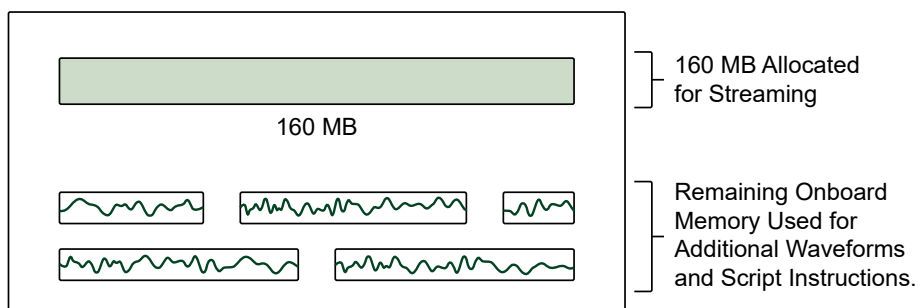
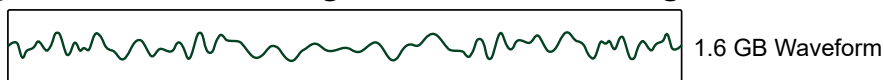
The following example shows a case when you want to generate a 1.6 GB waveform using an NI arbitrary waveform generator with 256 MB of onboard memory. This 1.6 GB waveform may be in the host memory, on disk, or data that your application generates dynamically during generation.



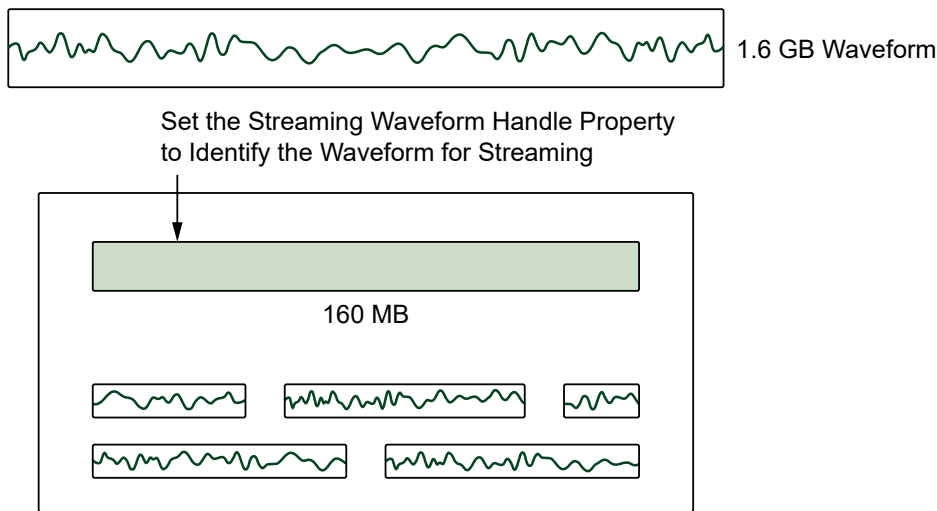
Complete the following steps to configure your application for streaming:

1. Call the `niFgen Allocate Waveform VI` or the `niFgen_AllocateWaveform` function to specify the amount of onboard memory to reserve for streaming.

The allocated memory, known as the ***streaming waveform***, serves as a buffer for the streaming process. The size of the waveform you want to stream must be evenly divisible by the amount of onboard memory allocated for streaming to prevent the streaming waveform from being overwritten before it has generated.

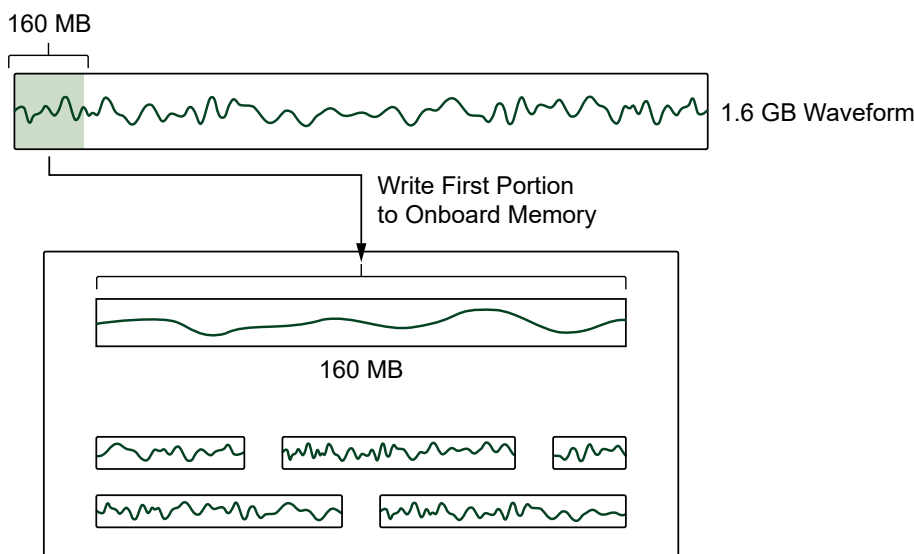


2. Set the `Streaming Waveform Handle` property or the `NIFGEN_ATTR_STREAMING_WAVEFORM_HANDLE` attribute to the waveform handle returned in step 1.
Setting this property or attribute ensures that none of your streaming data is overwritten before it is generated. NI-FGEN monitors your progress to ensure that you write fresh data fast enough to keep up with the generation. If your application fails to keep up or attempts to write fresh data over data that has not been generated, NI-FGEN returns an error.

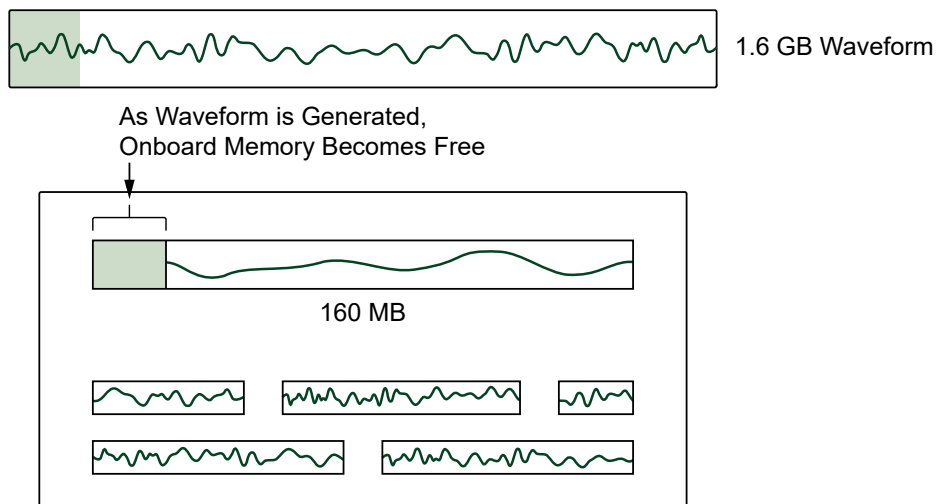


3. Call the niFgen Write Waveform VI or the niFgen_WriteWaveform function to write the first part of the waveform data to the streaming waveform in onboard memory.

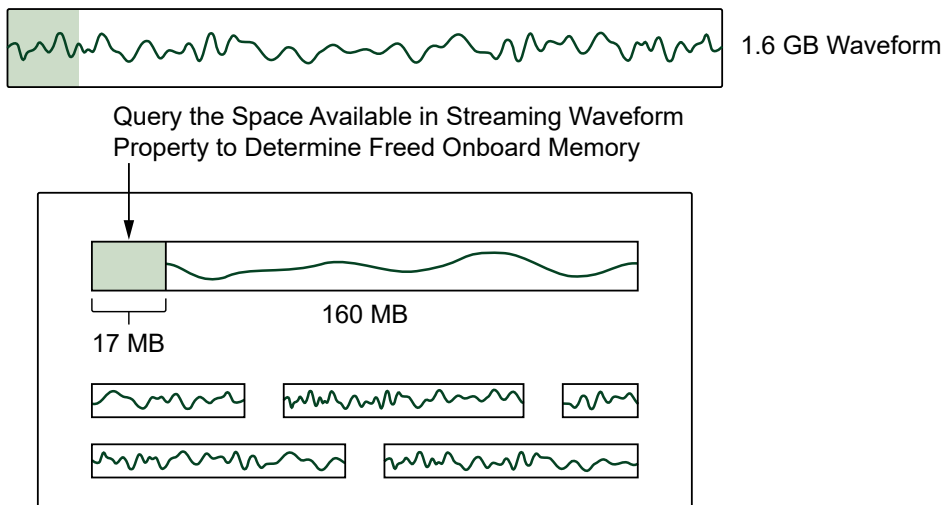
 **Tip** When transferring large blocks of waveform data, break the data into smaller blocks and call the niFgen Write Waveform VI or the niFgen_WriteWaveform function multiple times. The data is appended sequentially. A computer can allocate smaller blocks of a large waveform faster than allocating a single large contiguous block in memory. Depending on the amount of RAM on the computer, transferring ten 16 MB blocks may be faster than transferring one 160 MB block.



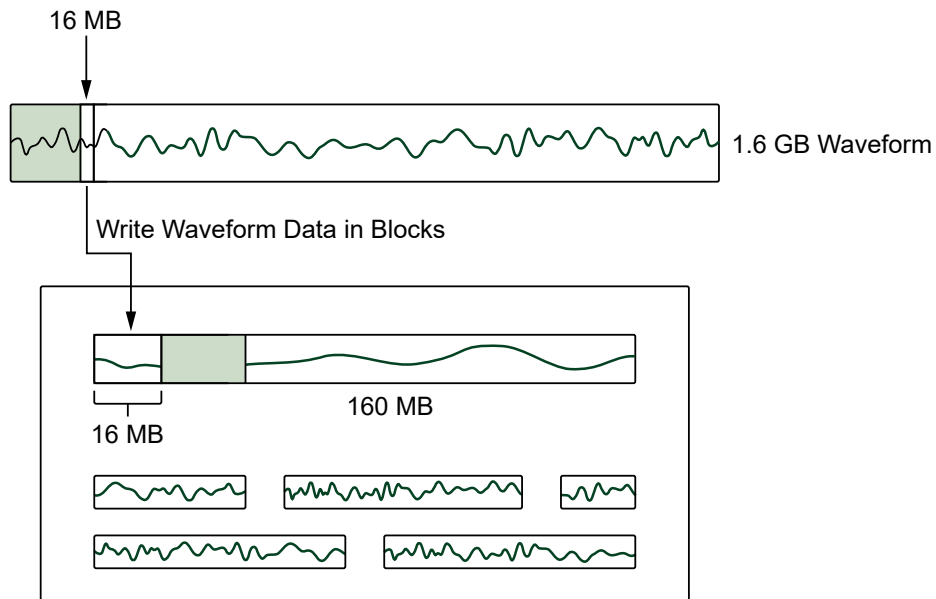
4. Call the niFgen Initiate Generation VI or the niFgen_InitiateGeneration function to begin the waveform generation.
As the waveform generates, space in the streaming waveform becomes free.



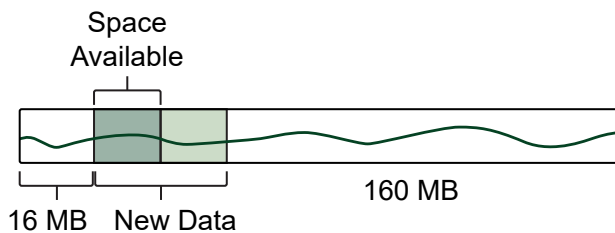
5. Use the Space Available in Streaming Waveform property or the `NIFGEN_ATTR_STREAMING_SPACE_AVAILABLE_IN_WAVEFORM` attribute to determine how much of the streaming waveform is free for writing new data. As the waveform generates, space becomes available to write more waveform data.



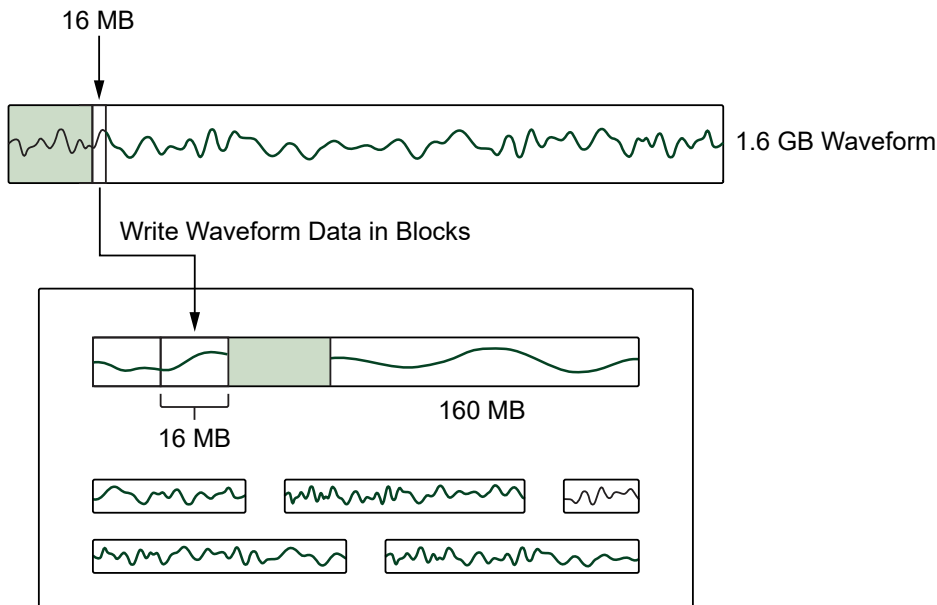
6. Call `niFgen Write Waveform VI` or the `niFgen_WriteWaveform` function to write a new block of waveform data to the streaming waveform in onboard memory.



If the size of the new block of waveform data is larger than the space available, NI-FGEN waits until sufficient space becomes available or the streaming write time expires. Use the `Streaming Write Timeout` property or the `NIFGEN_ATTR_STREAMING_WRITE_TIMEOUT` attribute to change the streaming write time.



7. Repeat steps 5 and 6 until all waveform data is written.



Streaming Waveform Data to Multiple Channels

To stream data to multiple channels, you must provide interleaved data. You cannot stream to two or more channels with individual, non-interleaved writes for each channel.

Improving Streaming Performance

To improve your maximum sustainable transfer rate for streaming, consider the following recommendations:

- Adjust the `Data Transfer Block Size` property or the `NIFGEN_ATTR_DATA_TRANSFER_BLOCK_SIZE` attribute. The default data transfer block size for NI-FGEN is 2 MSa (or 4 MB). If you were to write a 16 MB waveform to the signal generator, the complete transfer would occur using four separate DMA transfers. If you modify the data transfer block size to 8 MSa (16 MB), for example, the data transfer is more efficient and is instead accomplished in a single transfer.
- Configure advanced streaming properties by calling the `Maximum In-Flight Read Requests` or `Preferred Packet Size` property or the `NIFGEN_ATTR_DATA_TRANSFER_MAXIMUM_IN_FLIGHT_READS`, `NIFGEN_ATTR_DATA_TRANSFER_PCI_DMA_OPTIMIZATIONS`, or `NIFGEN_ATTR_DATA_TRANSFER_PREFERRED_PACKET_SIZE` attributes.

- Optimize the bus bandwidth usage for multi-device streaming applications by calling the `Maximum Bandwidth` property or the `NIFGEN_ATTR_DATA_TRANSFER_MAXIMUM_BANDWIDTH` attribute.
- When streaming from drives, consider the drive speed for maximum sustainable rates.

Transfer rates from drives can vary for a number of reasons, including where the data is physically stored on the drive and how much data is stored. Storing your waveform files on a fairly empty, defragmented drive may help increase performance.

- Use a redundant array of independent disks (RAID) configuration to utilize striping to increase data transfer rates from disk.
- When using 18-slot PXI chassis, install the signal generator used for streaming in the first segment (slots 2 to 6) of the PXI chassis.
- Utilize Direct DMA.

Using Direct DMA to Improve Streaming Performance

You can achieve high rates of data transfer to the onboard memory by configuring your device for Direct DMA.

1. Enable the signal generator for direct DMA writes by setting the `Direct DMA Enabled` property or the `NIFGEN_ATTR_DIRECT_DMA_ENABLED` attribute. After enabled, NI-FGEN monitors and reports any issues with the direct DMA transfer.
2. Identify the waveform data source and set the `Direct DMA Window Address` property or the `NIFGEN_ATTR_DIRECT_DMA_WINDOW_ADDRESS` attribute to the address provided by your direct DMA-compatible data source.
3. Set the `Direct DMA Window Size` property or the `NIFGEN_ATTR_DIRECT_DMA_WINDOW_SIZE` attribute to the size of the memory window provided by your direct DMA-compatible data source.
4. Use the `niFgen Write Waveform (Direct DMA I16)` VI or the `niFgen_WriteBinary16Waveform` function to write blocks of data to the signal generator. For each block of data written to the signal generator, you provide the address of the direct DMA window instead of an array of samples residing in host memory. NI-FGEN detects when the address is within the direct DMA window and handles the transfer appropriately.

Configuring Independent-Channel Waveform Generation

The PXle-5413 (two-channel) supports independent channels and therefore can be configured for multiple channel-based sessions or multichannel sessions.



Note Some settings are only instrument-based, as opposed to channel-based.

Configuring Multiple Channel-Based Sessions

All sessions open to the same waveform generator must use the same reference clock source and same setting for the auxiliary power pin (enabled/disabled).

1. Set the `Channel Name` parameters of two instances of the `niFgen Initialize With Channels VI` or `niFgen_InitializeWithChannels` function to different channels.
2. If you use the `Channel Name` parameter in the subsequent VI or function calls, set it to the same channel as specified with the `niFgen Initialize With Channels VI` or `niFgen_InitializeWithChannels` function.

Configuring Multichannel Sessions

1. Call any `niFgen Initialize VI` or function, or specify multiple channels using the **Channel Name** parameter of the `niFgen Initialize With Channels VI` or `niFgen_InitializeWithChannels` function or using the `Option String` parameter of any `Initialize VI` or function.
The `niFgen Initialize VI` and function use all channels on multichannel waveform generators by default and don't support reset when used in multichannel mode.
2. Provide interleaved data with the following:
 - The `niFgen Create Waveform VI` or `niFgen_CreateWaveform` functions
 - The `niFgen Write Waveform VI` or `niFgen_WriteWaveform` functions
 - The `niFgen Write Named Waveform VI` or `niFgen_WriteNamedWaveform` functions

Simulating the PXIe-5413

The PXIe-5413 supports simulation. To enable the simulation mode, set the `Option String` parameter of the `niFgen Initialize With Options VI` or the `niFgen_InitWithOptions` function as follows.

Model Variant	Option String Value
PXIe-5413 (One-Channel)	<code>Simulate=1,DriverSetup=Model:PXIe-5413(1CH)</code>
PXIe-5413 (Two-Channel)	<code>Simulate=1,DriverSetup=Model:PXIe-5413(2CH)</code>

Simulating an instrument enables you to perform the following tasks:

- Protect your instruments by first testing settings and configurations on simulated instruments.
- Verify instrument behaviors under a wide variety of operating conditions.
- Start or speed up application development before you have the hardware.
- Optimize designs and determine ideal design parameters.

Enabling simulation allows you to verify that you have correctly configured the instrument. For example, if a parameter is set to an invalid value for the instrument, NI-FGEN returns the same error it would for a real instrument.

While simulation is useful for verifying your configuration, there are some areas where simulation is not sufficient to verify that your configuration is correct. For example, no errors will be returned for configurations that involve or depend upon external signals, such as configuring an external sample clock or routing signals. Also, the amount of time a generation takes to complete will be ignored in simulation mode; a finite generation will finish immediately after it is initiated, regardless of how much data is downloaded and how fast it is generated.

Generating Waveforms

You can use InstrumentStudio or LabVIEW to generate a standard waveform or an arbitrary waveform.

Generating Waveforms with InstrumentStudio

You can use InstrumentStudio to generate a standard waveform or an arbitrary waveform.

1. Launch InstrumentStudio.
2. Select **File » New » Waveform Generator**.
3. In the instrument configuration panel, select the waveform mode from the drop-down menu and complete the following steps based on your selection:
 - **Standard waveform:**
 - a. Select the waveform type from the **Waveform** drop-down menu.



Note If you select **User-defined**, you need to upload the waveform file using the **Filename** box.

- b. Set frequency, amplitude, and offset.
 - c. Enable output for your channel.
 - d. If you are using a two-channel PXIe-5413, repeat previous steps for all the channels.
- **Arbitrary waveform:**
 - a. Specify the waveform file using the **Filename** box.
 - b. Set gain, offset, sample rate, and mode.
 - c. Enable output for your channel.
 - d. If you are using a two-channel PXIe-5413, repeat previous steps for all the channels.

4. Click **Run** to start waveform generation.

Generating Waveforms Using LabVIEW

1. Launch LabVIEW.
2. Select **Help » Find Examples**.

3. Select **Hardware Input and Output » Modular Instruments » NI-FGEN (Signal Generators) » Arbitrary Waveform Generation**.

4. Double-click an example to open it.

The `Fgen Basic Arb Waveform` example demonstrates how to get started using arbitrary waveform mode.

5. Follow the instructions on the front panel to run the example program.

You can refer to the steps on the block diagram to understand the example program.

Cleaning the PXIe-5413 System

NI recommends the following to clean and maintain your instrument's system:

- Clean the fan filters on the chassis regularly to prevent fan blockage and to ensure efficient air circulation. Cleaning frequency depends on the amount of use and the operating environment. For specific information about cleaning procedures and other recommended maintenance, refer to the chassis user documentation.
- Clean the hardware with a soft, nonmetallic brush. Make sure that the hardware is completely dry and free from contaminants before returning it to service.

PXIe-5413 Fundamental Concepts

Refer to the following topics to understand fundamental concepts about the waveform generators.

PXIe-5413 Bandwidth and Passband Flatness

The bandwidth of a signal source is defined as the frequency at which the amplitude of the frequency response is 3 dB lower than the amplitude of the frequency response at DC or a low frequency. The bandwidth of a source is limited by the output amplifier design or by filters in the analog output circuit. Bandwidth is one of the factors that determines the capability of the source to create signals with specific frequency content.

Passband flatness is a measure of the amplitude accuracy of the frequency response with respect to frequency. Passband flatness is usually specified in $\pm$ dB, and it is usually referenced to the amplitude of the frequency response at a designated frequency.

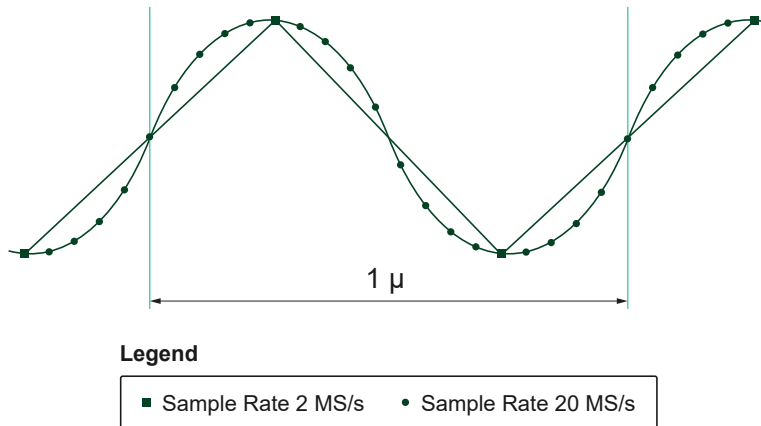
For example, a specification might be listed as ± 1 dB with respect to the amplitude of the frequency response at 50 kHz. This method is used because two different metrology instruments, a digital multimeter (DMM) and a power meter, are used to measure passband flatness. The power meter is an excellent metrology instrument for measuring passband flatness, but its performance can be improved by calibrating its frequency response at a low frequency, such as 50 kHz, with a DMM. In other words, the DMM measures the amplitude of a 50 kHz tone, and the power meter measures the amplitude of all other frequencies with respect to the amplitude of the 50 kHz tone measured by the DMM.

Passband flatness is important in many applications. For example, if the sensitivity of a receiver is being tested, it is important to know the variation of the amplitude of the test tone as it is swept across the frequency band of interest. Some NI signal generators have a Direct Output analog path that has been optimized for passband flatness. Other signal generators allow you to select a frequency at which the calibrated amplitudes can be finely adjusted to achieve the best amplitude accuracy near the selected frequency.

PXIe-5413 Sample Rate

Sample rate is the rate at which digital data is transferred from the memory to the digital-to-analog converter (DAC).

According to Shannon's Sampling theorem, a digital waveform must be updated at least twice as fast as the bandwidth of the signal to be accurately generated. Ideally, a sample rate many times greater than the frequency of the signal produces accurate waveforms. A higher sample rate also captures more waveform details. The following figure illustrates a 1 MHz sine wave generated by a sampled 2 MS/s DAC and a 20 MS/s DAC. The faster DAC generates 20 points per cycle of the expected signal compared with 2 points per cycle with the slower DAC. In this example, the higher sample rate more accurately defines the waveform shape.

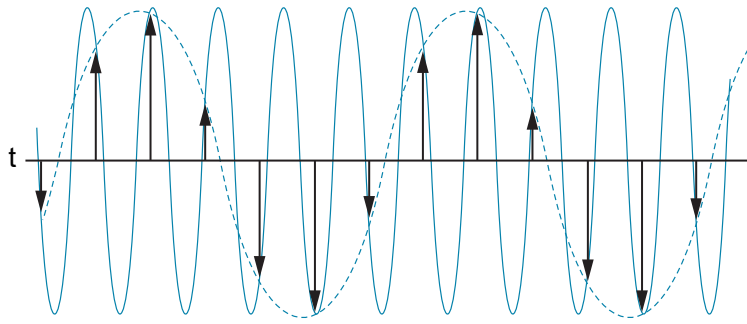


Nyquist and Shannon's Sampling Theorems

The Nyquist theorem concerns digital sampling of a continuous time analog waveform, while Shannon's Sampling theorem concerns the creation of a continuous time analog waveform from digital, discrete samples.

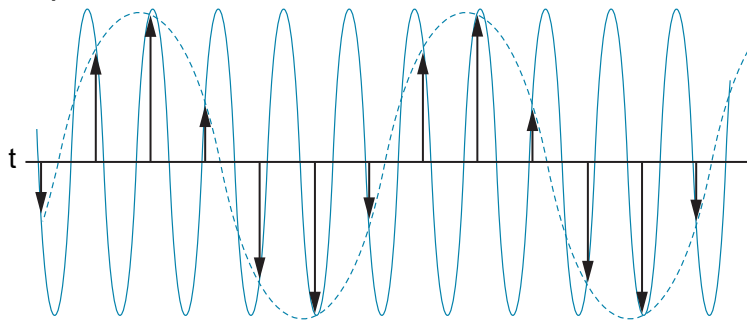
- **Nyquist Theorem**—The Nyquist theorem states that an analog signal must be sampled at least twice as fast as the bandwidth of the signal to accurately reconstruct the waveform; otherwise, the high-frequency content creates an alias at a frequency inside the spectrum of interest (passband). An alias is a false lower frequency component that appears in sampled data acquired at too low a sampling rate. The following figure shows a 5 MHz sine wave digitized by a 6 MS/s analog-to-digital converter (ADC). In this figure, the solid line represents the sine

wave being digitized, while the dotted line represents the aliased signal recorded by the ADC at that sample rate.



The 5 MHz frequency aliases back in the passband, falsely appearing as a 1 MHz sine wave.

- Shannon's Sampling Theorem**—Shannon's Sampling theorem states that a digital waveform must be updated at least twice as fast as the bandwidth of the signal to be accurately generated. The same image that was used for the Nyquist example can be used to demonstrate Shannon's Sampling theorem. The following figure shows a desired 5 MHz sine wave generated by a 6 MS/s DAC. The solid line represents the desired waveform, and the arrows represent the digitized samples that are available to recreate the continuous time 5 MHz sine wave. The dotted line indicates the signal that would be seen, for example, with an oscilloscope at the output of a DAC.



In this case, the high-frequency sine wave is the desired signal, but was severely undersampled by only being generated by a 6 MS/s DAC; the actual resulting waveform is a 1 MHz signal.

In systems where you want to generate accurate signals using sampled data, the sampling rate must be set high enough to prevent aliasing.

PXIe-5413 DAC Resolution

Digital-to-analog converter (DAC) resolution is a limiting factor in determining the accuracy of the re-creation of an analog waveform from digital samples. More details are present in the waveform if the DAC resolution is increased. A 3-bit DAC divides the vertical range into eight discrete levels. With a vertical range of 10 V, the 3-bit DAC cannot generate voltage differences smaller than 1.25 V. In comparison, a 16-bit DAC with 65,536 discrete levels can generate voltage differences as small as 153 μV .

The following figure shows the difference between two waveforms. The 16-bit waveform looks like a continuous sine wave, but if you were to zoom in, you would see the discrete steps of 153 μV . Both waveforms are composed of discrete voltage steps, but the 16-bit version looks much closer to a "pure" continuous-time sine waveform.

