

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com

**Full-service, independent repair center**
with experienced engineers and technicians on staff.

**We buy your excess, underutilized, and idle equipment**
along with credit for buybacks and trade-ins.

**Custom engineering**
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

**Expert team | Trust guarantee | 100% satisfaction**

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

**Visit our website - *Click HERE***

# PXIe-6571 User Manual

2025-04-15

# Contents

Welcome to the PXIe-6571 User Manual . . . . . 4
PXIe-6571 Overview . . . . . 5
Components of a PXIe-6571 System . . . . . 8
PXIe-6571 Theory of Operation . . . . . 15
PXIe-6571 Front Panel . . . . . 17
PXIe-6571 Pinout . . . . . 19
PXIe-6571 LED Indicators . . . . . 23
PXIe-6571 Safety Guidelines . . . . . 26
PXIe-6571 Installation and Configuration . . . . . 28
Unpacking the Kit . . . . . 28
Installing the Software . . . . . 29
Installing the PXIe-6571 into a Chassis . . . . . 30
Verifying the Installation . . . . . 31
Self-Calibrating the PXIe-6571 . . . . . 34
Connecting Signals to the PXIe-6571 . . . . . 36
Connecting to a Device Under Test . . . . . 36
Installing the CB-2162 on the PXIe-6571 . . . . . 36
Installing the SMB-2163 on the PXIe-6571 . . . . . 39
Installing the SCB-68 HSDIO on the PXIe-6571 . . . . . 42
Installing the SHC68-H1X38 . . . . . 44
Triggers and Events . . . . . 46
Signal Routing . . . . . 48
Synchronizing Multiple Instruments . . . . . 50
Making Frequency Measurements . . . . . 54
Correctly Measuring Signal Integrity . . . . . 56
Sourcing and Capturing Waveforms . . . . . 57
Scan Testing . . . . . 61
Example Programs . . . . . 63
PXIe-6571 Operating Guidelines . . . . . 65
Pin Electronics . . . . . 65
Programming Pin Electronics . . . . . 66
Programming Pin Driver . . . . . 70

Programming Comparator . . . . . 72
Programming Active Load . . . . . 73
Programming PPMU . . . . . 74
Clocking . . . . . 82
Clock Generator . . . . . 83
Sequencer Flags and Registers . . . . . 84
Vector Memory . . . . . 84
Edge Multiplier . . . . . 85
Shared Pins . . . . . 86
TDR . . . . . 87
History RAM . . . . . 88
Thermal Protection . . . . . 89
Calibration . . . . . 90

# Welcome to the PXIe-6571 User Manual

The PXIe-6571 User Manual provides detailed descriptions of product functionality and 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-6571 Specifications
- PXIe-6571 Calibration Procedure
- NI-Digital Pattern Driver Python API Documentation
- Digital Pattern User Manual
- NI-TClk Synchronization Help
- PXIe-6674T User Manual
- Software and Driver Downloads
- Using an Info Code
- Calibration Services
- NI Package Manager Manual
- PXI Slot Blocker Documentation
- NI Learning Center
- Discussion Forums
- Letter of Volatility
- Product Certifications
- Dimensional Drawings
- License Setup and Activation
- Release Notes

# PXIe-6571 Overview

The PXIe-6571 is a 1-slot, 100 MVector/s digital pattern instrument, available in 8-channel and 32-channel variants, that features enhanced capabilities for semiconductor characterization, production test, and validation test; it is designed for semiconductor test engineers in digital testing. Use the PXIe-6571 in applications including digital and mixed signal systems functional test, specification characterization for timing and voltage, and parametric measurement tests.



**Note** Unless otherwise noted, "PXIe-6571" encompasses both the 8-channel and 32-channel variants.

## Device Capabilities

The PXIe-6571 has the following features and capabilities.

- 100 MVector/s PXI Digital Pattern Instrument
- 100 MHz vector rate
- System channel count[1] up to 512 for the PXIe-6571 (32-channel)
- Pattern formats including non-return, return to low, return to high, surround by complement
- 31 time sets for pattern timing
- 128 M per channel vector memory depth
- Supports multisite testing, up to 8 sites per digital pattern instrument
- Resources for 8 source and 8 capture sites

## Driver Support

NI recommends that you use the newest version of the driver for your module.

The driver for the PXIe-6571 is NI-Digital Pattern Driver.

1. The ***system channel count*** is the maximum number of channels available when using multiple PXIe-6571 (32-channel) instruments in a single chassis as a digital subsystem within an application system.

**Table 1.** Earliest NI-Digital Pattern Driver Version Support

| Module | Earliest Version Support |
|---|---|
| PXIe-6571 (8-channel) | 2024 Q2 |
| PXIe-6571 (32-channel) | 18.0 |

# Distinguishing PXIe-6571 Variants

You can differentiate the 8-channel and 32-channel PXIe-6571 within NI configuration software, programmatically with the System Configuration API, and visually.

In general, the 8-channel instrument is designated PXIe-6571 (8CH) and the 32-channel instrument is designated PXIe-6571.

Distinguish between the 8-channel and 32-channel PXIe-6571 instruments as follows, depending on the method.

| Option | Description |
|---|---|
| **Configuration software: Hardware Configuration Utility, MAX** | View the instrument within NI hardware configuration software.<br>• 8-channel: PXIe-6571 (8CH)<br>• 32-channel: PXIe-6571 |
| **Programmatic: System Configuration API** | Build a program to identify hardware, such as the Show All Hardware example.<br>• 8-channel: PXIe-6571 (8CH)<br>• 32-channel: PXIe-6571 |
| **Visual** | • **Front panel**—View the text next to the front panel VHDCI connector.<br>  • 8-channel: 8-CH DIGITAL DATA AND CONTROL<br>  • 32-channel: DIGITAL DATA AND CONTROL<br>• **Module label**—View the label on the left side of the module, relative to viewing the front panel, to find the assembly part number. |

| Option | Description |
| --- | --- |
| | • 8-channel: 147558***X***-02L, where ***X*** is a letter<br>• 32-channel: 147558***X***-01L, where ***X*** is a letter |

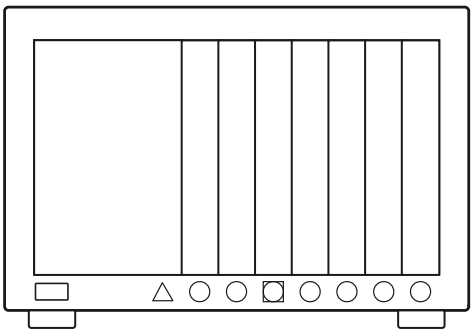
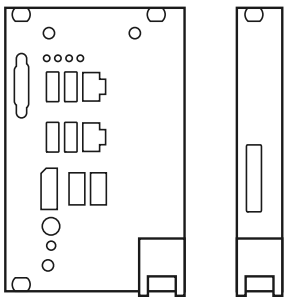
# Components of a PXIe-6571 System

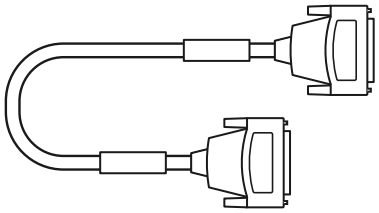
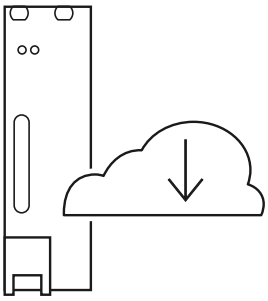
The PXIe-6571 is designed for use in a system that includes other hardware components, drivers, and software.

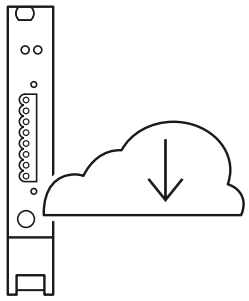
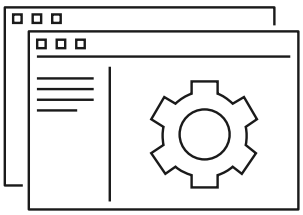
**Notice** A system and the surrounding environment must meet the requirements defined in ***PXIe-6571 Specifications***.

The following list defines the minimum required hardware and software for a system that includes a PXIe-6571.

**Table 2.** System Components

| Component | Description and Recommendations |
|---|---|
| PXI Chassis  | A PXI chassis houses the PXIe-6571 and supplies power, communication, and timing for PXIe-6571 functions.<br>**Notice** The 32-channel PXIe-6571 requires an 82 W power and cooling–capable chassis; the 8-channel PXIe-6571 requires a ≥58 W power and cooling–capable chassis. Using the PXIe-6571 in a chassis with lower power or cooling capabilities than required will result in an error. |
| PXI Controller or PXI Remote Control Module  | You can install a PXI controller or a PXI remote control (MXI) module depending on your system requirements. These components, installed in the same PXI chassis as the PXIe-6571, interface with the instrument using NI instrument drivers. |

| Component | Description and Recommendations |
|---|---|
| | |
| Digital Pattern Instrument  | Your digital pattern instrument. |
| Cables and Accessories  | Cables and accessories allow connectivity to/from your instrument for measurements. Refer to ***Cables and Accessories*** for recommended cables and accessories and guidance. |
| NI-Digital Pattern Driver  | Instrument driver software that provides functions to interact with the PXIe-6571 and execute measurements using the PXIe-6571.<br>**Note** NI recommends that you always use the most current version of NI-Digital Pattern Driver with the PXIe-6571. |
| Additional NI Drivers | • **NI-DCPower**—The NI-DCPower driver is required to use NI-DCPower modules within the Digital Pattern Editor.<br>• **NI-Sync**—The NI-Sync driver is required to support operations using the PXIe-6674T timing and |

| Component | Description and Recommendations |
|---|---|
|  | synchronization module, such as combining comparison results across digital pattern instruments.<br>For more information about using NI-DCPower modules with the Digital Pattern Editor and using the PXIe-6674T with digital pattern instruments, refer to the ***Digital Pattern User Manual***. |
| NI Applications<br> | NI-Digital Pattern Driver offers driver support for the following applications:<br>• InstrumentStudio<br>• LabVIEW<br>• TestStand<br>• TestStand Semiconductor Module™<br>• C/C++<br>• .NET<br>• Python |

# PXIe-6571 Cables and Accessories

NI recommends using the following cables and accessories with your module.

| Accessory Description | Notes | NI Part Number |
|---|---|---|
| CB-2162 Connector Block and Prototyping Board | DIO channels: 32 | 778592-01 |
| SCB-68 HSDIO Shielded Connector Block for R Series and HSDIO Devices | DIO channels: 32 | 782914-01 |
| SMB-2163 SMB Breakout Box for High-Speed Digital Devices | DIO channels: 32 | 778747-01 |

| Accessory Description | Notes | | NI Part Number |
|---|---|---|---|
| NI SHC68-C68-D4 Shielded Single-Ended Cable for High-Speed Digital Devices | DIO channels: 40 | 0.55 m length | 781013-01 |
| | | 1 m length | 196275-01 |
| | | 1 m length, low leakage | 152870-01 |
| | | 2 m length | 781293-01 |
| SHC68-H1X38 High-Speed Digital Flying-Lead Cable, 1.5 m | DIO channels: 32 | | 192681-1R5 |
| Single Stack Right Angle Through Hole Mount 68-pin VHDCI Connector/<br><br>Double Stack Right Angle Through Hole Mount 68-pin VHDCI Connector/<br><br>Vertical Through Hole Mount 68-pin VHDCI Connector | — | | 785633-01/<br><br>785634-01/<br><br>785753-01 |
| PXI slot blockers | — | | 199198-01 |

**Note** For more information about recommended accessories for use within a system such as STS, contact your NI Sales Engineer.

## Additional Cabling and Accessory Guidance

- NI recommends installing PXI slot blockers (NI part number 199198-01) to fill empty instrument slots in a PXI chassis. For more information about installing slot blockers and filler panels, refer to PXI slot blocker documentation.

**Related information:**

- Right Angle VHDCI Connectors Replacement Notice

- Vertical VHDCI Connector Replacement Notice
- PXI Slot Blocker Documentation

## Programming Options

You can use Digital Pattern Editor to operate your digital pattern instrument or you can use the NI-Digital Pattern Driver to program your instrument in the supported ADE of your choice.

- **Digital Pattern Editor**—Use Digital Pattern Editor to view, create, modify, and debug pin and channel maps, specifications, levels, timing, patterns, register maps, source waveforms, and capture waveforms. You can also use Digital Pattern Editor to configure the state of the digital pattern instrument.

  The Digital Pattern Editor is automatically installed when you install NI-Digital Pattern Driver. You can access the Digital Pattern Editor by selecting **National Instruments** » **NI Digital Pattern Editor** from the Windows start menu.

- **NI-Digital Pattern Driver**—Use the NI-Digital Pattern Driver APIs to call pin and channel map, specifications, levels, timing, pattern, and waveform files you create in the Digital Pattern Editor and to configure and control digital pattern devices.
  - LabVIEW—Available on the LabVIEW Functions palette at **Functions** » **Instrument I/O** » **Instrument Drivers** » **NI-Digital palette** or at **Functions** » **Measurement I/O** » **NI-Digital palette** .

    Examples are available from the **Start** menu in the **National Instruments** folder.

    The NI-Digital Pattern Driver LabVIEW API is installed by default when you run the installer.

  - C—You can use the NI-Digital Pattern C dynamically linked library by adding a reference to `C:\Program Files (x86)\IVI Foundation\IVI\Bin\niDigital_32.dll` for 32-bit development or to `C:\Program Files\IVI Foundation\IVI\Bin\niDigital_64.dll` for 64-bit development. The NI-Digital Pattern Driver API is installed by default when you run the installer.
  - .NET—Use the NI-Digital Pattern .NET class library by adding a reference to

NationalInstruments.ModularInstruments.NIDigital.Fx40 or NationalInstruments.ModularInstruments.NIDigital.Fx45 and any dependent class libraries from within the Solution Explorer in Microsoft Visual Studio.

You can optionally install and use the NI-Digital Pattern Driver .NET API to configure and control the digital pattern instrument.

The Microsoft .NET examples are available from the **Start** menu in the **National Instruments** » **NI Digital Pattern Examples** folder or from the `<Public Documents>\National Instruments\NI-Digital\Examples\DotNET 4.x` directory.

- Python—For more information about installing and using Python, refer to ***NI-Digital Pattern Driver Python API Documentation***.

## Chassis Guidelines

Digital pattern instruments are designed to operate in a PXI Express chassis with a well-designed cooling system at specified environmental conditions.

**Note** The 32-channel PXIe-6571 is compatible only with 82 W power and cooling–capable chassis; the 8-channel PXIe-6571 is compatible with ≥58 W power and cooling–capable chassis.

Instrument performance and reliability might be limited at temperatures outside of the specified operating range, and operating under high humidity or dusty conditions can cause leakages among circuit components to increase and might result in additional measurement errors.

To guarantee that the instrument meets its listed specifications, ensure that the ambient temperature is within the temperature range listed in the instrument specifications and that the temperature is stable at $T_{cal}$ ±5 °C. $T_{cal}$ is the internal instrument temperature recorded by the digital pattern instrument at the completion of the last self-calibration.

## Chassis Cooling Guidelines

Use the following guidelines to optimize cooling and ensure high-quality performance and reliability:

- Fill all empty slots with PXI slot blockers.
- Cover all empty slots in the chassis with an EMC slot filler panel.
- Remove and clean the inlet filters often to prevent buildup of dust and other foreign material that might restrict airflow.
- Position the chassis such that the fan inlets and outlet vents are not obstructed. Keep other objects and equipment a minimum of 4 inches away from the fan inlets.

For more information about cooling considerations, refer to your chassis documentation.

## Digital Pattern Instrument Placement in Chassis

Digital pattern instruments are high-precision modules and might be sensitive to interference from other electronic devices. To optimize the accuracy and performance of the digital pattern instrument, position the instrument in a slot away from third-party instruments that may include noisy circuitry, such as power supplies. The digital pattern instrument might also be sensitive to excess heat generated by high-power products in neighboring slots.

# PXIe-6571 Theory of Operation

The PXIe-6571 combines a high-performance digital backend with pin electronics in front to drive, compare, and measure signals to and from a device under test (DUT).

The pin electronics provide the interface for the module to the DUT pins. Each one can drive a high or low logic level at programmable voltages and can compare output levels to both a high and low logic threshold. The timing and voltages are configurable via the host interface.

Groups of pins are connected to a Pin Parametric Measurement Unit (PPMU) that provides the ability to force and measure current or voltage on pins, but not on all pins simultaneously.

The digital backend features an FPGA that provides the processing and logic for the module. The FPGA is connected to the PXIe backplane via a PCIe Gen2x4 link to provide configuration and data movement to and from the host and module. The onboard memory stores the vectors that define the drive and compare values and timing for each of the pins.

If the DUT pins are within the pin number limits of one instrument, you can test multiple DUTs simultaneously without manually duplicating pin values using multiple sites, one site per DUT.

If the DUT pins exceed the maximum pin number of one instrument, the PXIe-6571 connects to a PXIe-6674T via the backplane trigger bus to combine the pass or fail results across multiple instruments.

## PXIe-6571 Block Diagrams

The following diagrams illustrate the design of the PXIe-6571.

**Figure 1.** PXIe-6571 High-Level Block Diagram

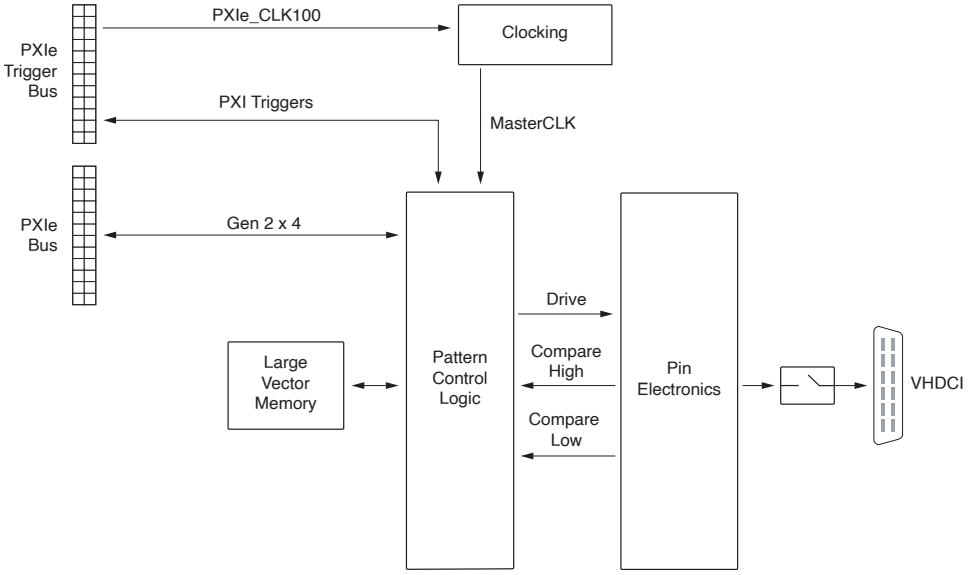
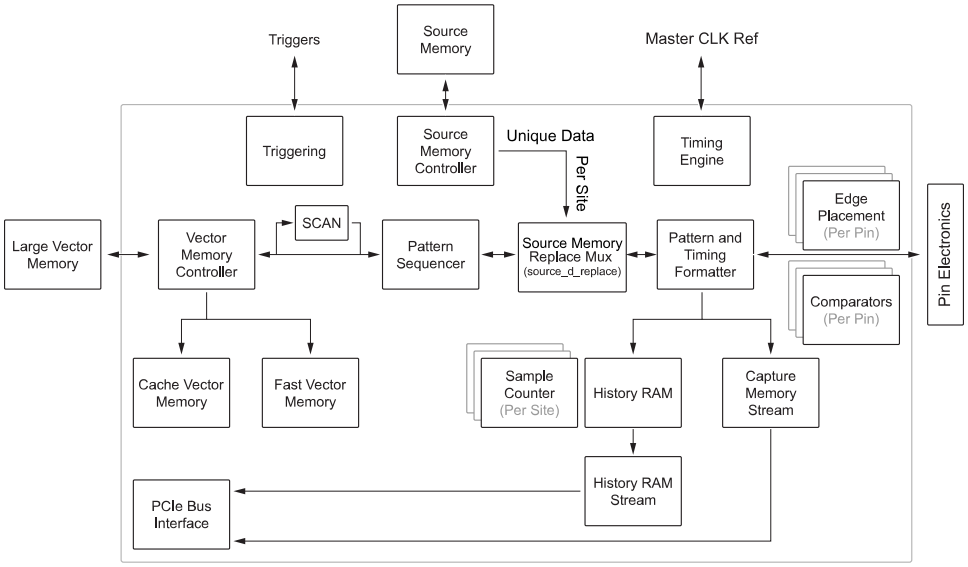


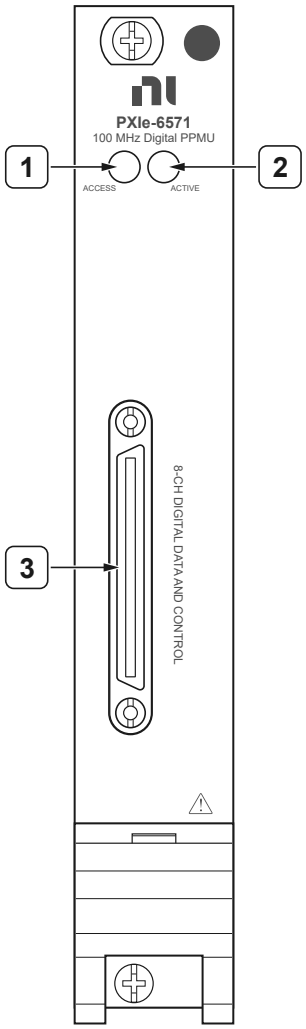

**Figure 2.** PXIe-6571 Pattern Control Logic Block Diagram



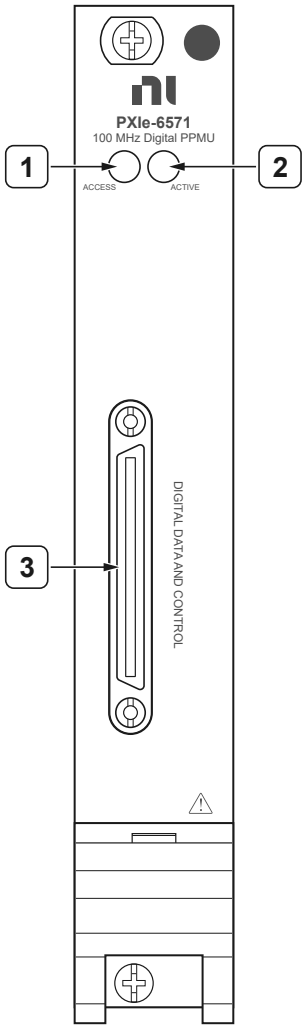
# PXIe-6571 Front Panel

**Figure 3.** PXIe-6571 Front Panel

PXIe-6571 (8-channel)




PXIe-6571 (32-channel)




1. Access LED
2. Active LED
3. 68-Pin VHDCI Digital Data and Control (DDC) Connector

# PXIe-6571 Pinout

The PXIe-6571 exposes signal terminals via a VHDCI connector.

**Figure 4.** PXIe-6571 (8-channel) Connector Pinout

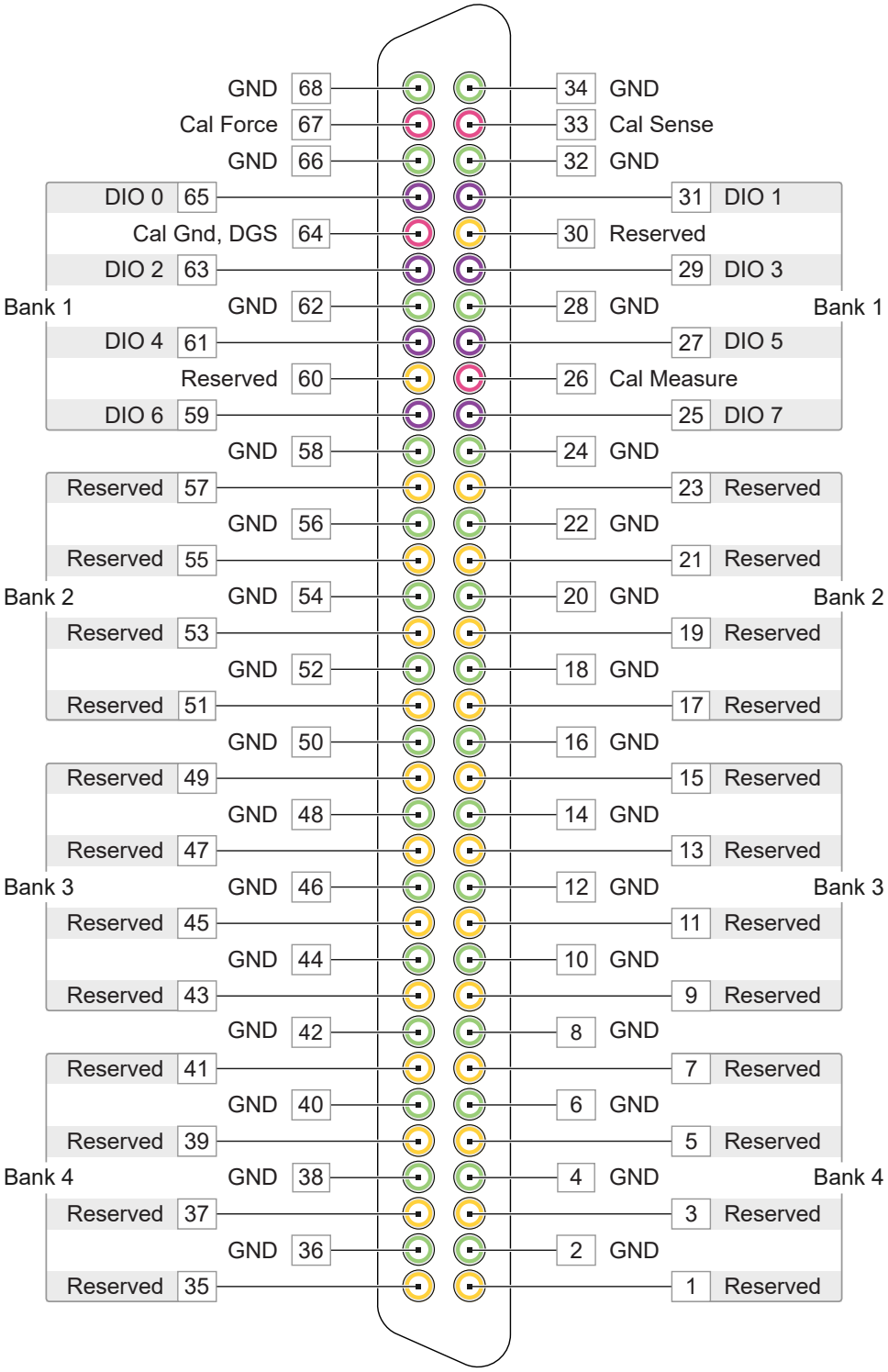


**Figure 5.** PXIe-6571 (32-channel) Connector Pinout

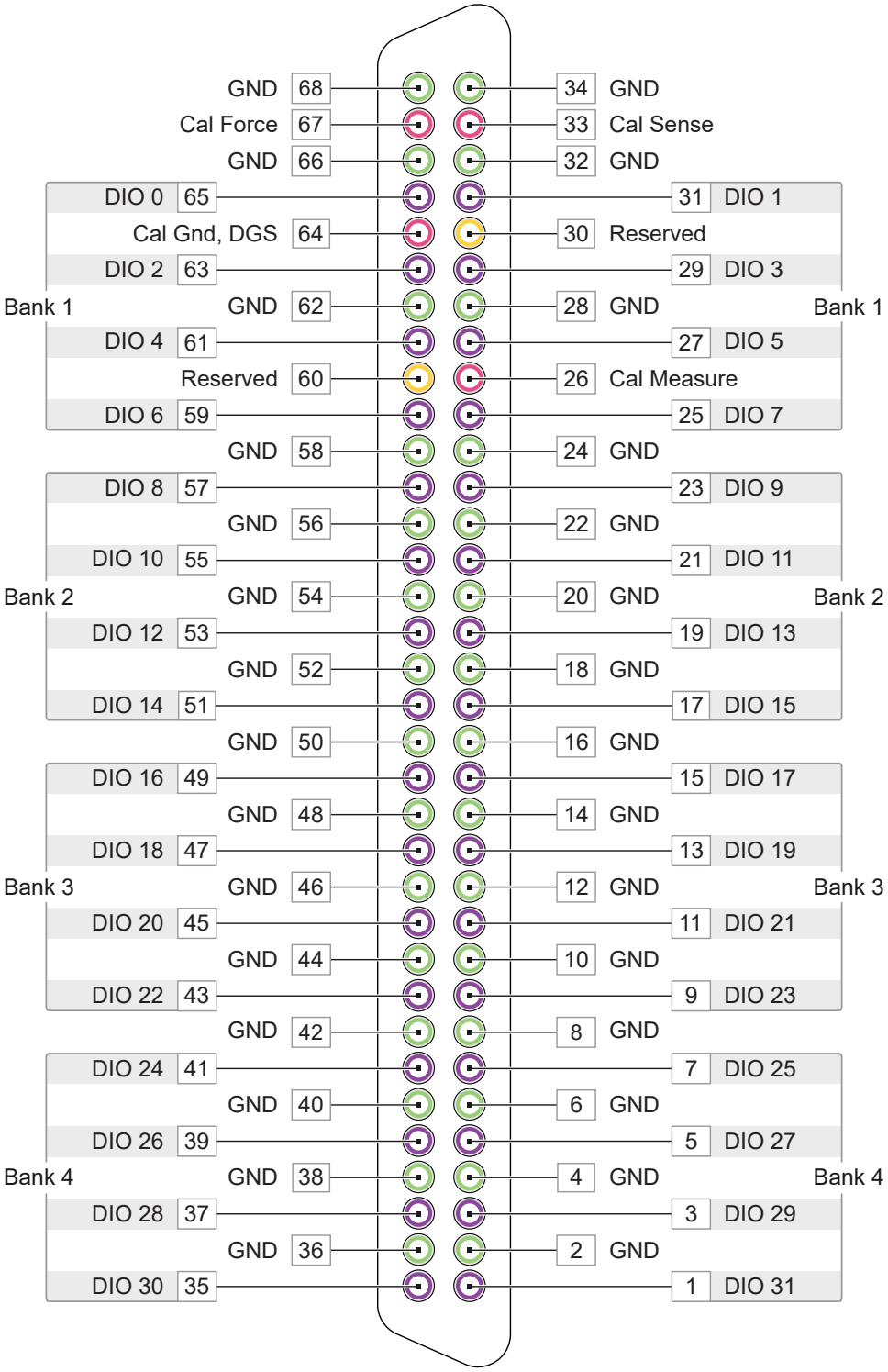


**Table 3.** PXIe-6571 Digital Data and Control Connector Pins/Signal Descriptions

| Signal Type | Signal Name | Signal Description |
|---|---|---|
| Data | DIO <0..31> | Bidirectional PPMU-capable digital I/O data channels 0 through 31. |
| Ground | GND | Instrument ground. Acts as the default ground reference when DUT Ground Sense (DGS) is not connected. |
| | DGS | Optional DGS for improved accuracy at higher currents in some configurations. |
| Analog | CAL MEASURE | Resource for external calibration. |
| | CAL SENSE | |
| | CAL GND | |
| | CAL FORCE | |
| N/A | RESERVED | These terminals are reserved for future use. Do not connect to these pins. |

**Note** The digital I/O data channels of 32-channel digital pattern instruments are split into banks for PPMU operation efficiency: DIO <0..7>, DIO <8..15>, DIO <16..23>, DIO <24..31>. PPMU measurements run in parallel when you take measurements on channels across different banks. Taking PPMU measurements simultaneously with channels on the same bank impacts test time performance based on certain measurement settings. Test time performance for frequency counter measurements is not impacted by taking multiple frequency counter measurements on channels in the same bank.

# PXIe-6571 LED Indicators

The PXIe-6571 features an Access LED and Active LED.

## Access LED

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

The following table lists the Access LED states.

**Table 4.** Access LED Indicator Status

| Status Indicator | Device State |
|---|---|
| Off | Device is not yet functional. |
| Amber | Device is being accessed by software. |
| Green | Device is 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 PXIe-6571 appears in Hardware Configuration Utility or MAX.

If 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 while the PXIe-6571 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.

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

> **Notice** Do not reinstall a damaged module.

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

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

6. Verify that the module appears in Hardware Configuration Utility or MAX.
7. Reset the module in Hardware Configuration Utility or MAX and perform a self-test.

## Active LED

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

The following table lists the Access LED states.

**Table 5.** Active LED Status Indicator

| Indicator Status | Indications |
|---|---|
| Off | Device is either in PPMU operation or not waiting for a trigger, not bursting a pattern, and not experiencing an error. |
| Amber | Device is awaiting the Start trigger. |
| Green | Device received the Start trigger and is bursting a pattern.<br>**Note** If self-test, self-calibration, or external calibration on the device fails, the LED remains green and you will receive an error. The error persists until self-test, self-calibration, or external calibration succeeds or the device is reset. |
| Red | Device is in error condition. |

## 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 device is reset.

| Possible Error | Solution |
|---|---|
| The instrument has detected an unlocked condition on a PLL that might result in incorrect data. | To clear this error, reset the instrument programmatically or through Hardware Configuration Utility or MAX. |
| The instrument has been disabled because it exceeded its overall power limit. | To re-enable the instrument, reset it programmatically or through Hardware Configuration Utility or MAX. |
| 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. Reset the instrument programmatically or through Hardware Configuration Utility or MAX, or power cycle the instrument. |

# 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.

**Related information:**

- Product Certifications

## Safety Voltages

Connect only voltages that are within these limits.

| | |
|---|---|
| Supported measurement range[2] | -2 V to 7 V[3] |
| Measurement Category | CAT I |

## Measurement Category

**Caution** Do not connect the product to signals or use for measurements within Measurement Categories II, III, or IV.

2. If the total voltage sourced or driven on any pin relative to GND exceeds the supported measurement range, instrument performance may be degraded.
3. ***Voltage*** > 6 V requires the Extended Voltage Range mode of operation.

**Attention** Ne pas connecter le produit à des signaux dans les catégories de mesure II, III ou IV et ne pas l'utiliser pour effectuer des mesures dans ces catégories.

Measurement Category I is for measurements performed on circuits not directly connected to the electrical distribution system referred to as ***MAINS*** voltage. MAINS is a hazardous live electrical supply system that powers equipment. This category is for measurements of voltages from specially protected secondary circuits. Such voltage measurements include signal levels, special equipment, limited-energy parts of equipment, circuits powered by regulated low-voltage sources, and electronics.

**Note** Measurement Categories CAT I and CAT O are equivalent. These test and measurement circuits are for other circuits not intended for direct connection to the MAINS building installations of Measurement Categories CAT II, CAT III, or CAT IV.

# PXIe-6571 Installation and Configuration

Complete the following steps to install the PXIe-6571 into a chassis and prepare it for use.

1. Unpacking the Kit
2. Installing the Software
3. Installing the PXIe-6571 into a Chassis
4. Verifying the Installation
   Verify that the PXIe-6571 is installed correctly by checking that Hardware Configuration Utility or MAX recognizes the instrument.
5. Self-Calibrating the PXIe-6571

## Unpacking the Kit

**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.

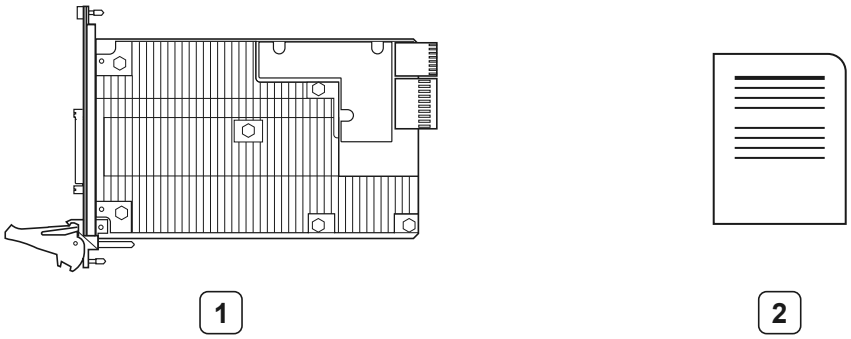
   **Note** Do not install a device if it appears damaged in any way.
3. Unpack any other items and documentation from the kit.

**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-6571 kit.

**Figure 6.** PXIe-6571 Kit Contents




1. PXIe-6571 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 Microsoft Visual Studio.
2. Download the driver software at ni.com/downloads.
   NI Package Manager downloads with the driver software to handle the installation.
   Refer to the ***NI Package Manager Manual*** for more information about installing, removing, and upgrading NI software using NI 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. Select **.NET Framework 4.0 Languages Support** or **.NET Framework 4.5 Languages Support** in the NI-Digital Pattern Driver installer to install .NET support for the NI-Digital Pattern Driver software.
5. When the installer completes, select **Restart** in the dialog box that prompts you to restart, shut down, or restart later.

# Installing the PXIe-6571 into a Chassis

**Notice** The 32-channel PXIe-6571 is compatible only with 82 W power and cooling–capable chassis; the 8-channel PXIe-6571 is compatible with ≥58 W power and cooling–capable chassis. Using the PXIe-6571 in a chassis with lower power or cooling capabilities than required will result in an error.

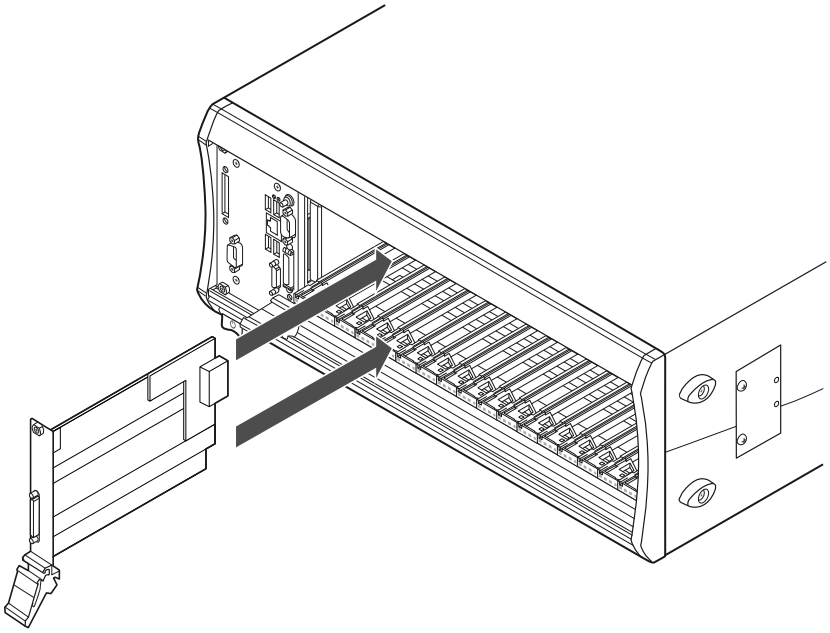
1. Ensure the AC power source is connected to the chassis before installing the module.
   The AC power cord grounds the chassis and protects it from electrical damage while you install the module.
2. Power off the chassis.
3. Inspect the slot pins on the chassis backplane for any bends or damage prior to installation. Do not install a module if the backplane is damaged.
4. Remove the black plastic covers from all the captive screws on the module front panel.
5. Identify a supported slot in the chassis.

   The PXIe-6571 module must be installed in a PXI Express-compatible slot; it can be placed in PXI Express hybrid peripheral slots (●$^{H}$) or PXI Express peripheral slots (●).

   **Tip** If you plan to use the optional PXIe-6674T Timing and Synchronization Module, reserve the PXI Express system timing slots (▣) for its placement in the chassis.

6. Touch any metal part of the chassis to discharge static electricity.
7. Ensure that the ejector handle is in the downward (unlatched) position.

**Figure 7.** Module Installation



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

> **Note** Tightening the top and bottom mounting screws increases mechanical stability and also electrically connects the front panel to the chassis, which can improve the signal quality and electromagnetic performance.

11. Cover all empty slots using either filler panels (standard or EMC) or slot blockers with filler panels, depending on your application.

> **Note** For more information about installing slot blockers and filler panels, go to ni.com/r/pxiblocker.

**Related information:**

- ni.com/r/pxiblocker

# Verifying the Installation

Verify that the PXIe-6571 is installed correctly by checking that Hardware Configuration Utility or MAX recognizes the instrument.

## Verifying the Installation in Hardware Configuration Utility

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

1. Open Hardware Configuration Utility.
   The PXIe-6571 should appear in the system pane automatically.
   If the PXIe-6571 does not appear automatically, do the following:
   - Click the **+** button or, in the menu, select **Edit » Add hardware...**.
   - Find your instrument in the Discovered pane and click **Add**.
2. Record the name Hardware Configuration Utility assigns to the PXIe-6571 or, if desired, provide a custom name to the PXIe-6571.
   Use this name when programming the PXIe-6571.
3. Validate that your instrument is installed correctly: select the PXIe-6571 module in the system pane, expand the **Troubleshooting** area of the configuration pane, and click **Self-test**.
   Hardware Configuration Utility reports when the hardware setup is validated.

## Verifying the Installation in MAX

You can 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-Digital Pattern Driver.



**Note** MAX is not available on Linux.

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 name MAX assigns to the hardware. Use this identifier when programming the PXIe-6571.
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.

## What Should I Do if 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 needs to be refreshed.

| Software | Description |
|---|---|
| **Hardware Configuration Utility** | Click the refresh button (). |
| **MAX** | a. In the MAX configuration tree, expand **Devices and Interfaces**.<br>b. Expand the **Chassis** tree to see the list of installed hardware and 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 by right-clicking the **Start** button and selecting **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<br>b. If error conditions appear in the list, right-click the module you are using in the |

| Controller Type | Description |
|---|---|
| | **Device Manager** and select **Update Driver**. |
| **MXI controller** | Right-click **PCI-to-PCI Bridge** and select **Properties** from the shortcut menu to verify that the bridge is enabled and that no error conditions appear. |

If error conditions persist, reinstall NI-Digital Pattern Driver.

6. Restart your computer.

## What Should I Do if the PXIe-6571 Fails the Self-Test?

1. Reset the PXIe-6571 through Hardware Configuration Utility or MAX and then perform the self-test again.
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 PXIe-6571

Self-calibration adjusts the PXIe-6571 for variations in the module environment. The PXIe-6571 modules are externally calibrated at the factory, however you should perform a complete self-calibration after you install the module.

1. Install the PXIe-6571 and let it warm up for the recommended warm-up time listed in the ***PXIe-6571 Specifications***.

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

2. Self-calibrate the PXIe-6571.

| Option | Description |
| --- | --- |
| **Interactive** | Click the **Self-Calibrate** button in Hardware Configuration Utility or MAX |
| **Programmatic** | Call the relevant NI-Digital Pattern Driver function in your program:<br>3. **LabVIEW**—Self Calibrate VI<br>4. **.NET/C#**—NIDigital.SelfCalibrate method<br>5. **C**—niDigital_SelfCalibrate function |

## Related information:

- PXIe-6571 Specifications

# Connecting Signals to the PXIe-6571

Refer to the following topics for guidance about PXIe-6571 signal connections.

## Connecting to a Device Under Test

To connect a device under test (DUT) to the PXIe-6571, create a custom device interface board to access the signals of the DUT and use compatible cabling between the PXIe-6571 and the device interface board.

For more information, refer to the ***Interfacing to the Digital Pattern Instrument or Digital Waveform Instrument using the VHDCI Connector*** application note. Go to ni.com/info and enter the Info Code `rdinwa` to locate the application note.

**Note** If you plan to use the PXIe-6571 as part of an integrated system, such as a Semiconductor Test System (STS), refer to the system documentation for guidance on device interface board design and connection guidelines. For system documentation, visit ni.com/docs or contact your NI sales engineer.

To connect a custom device interface board to the VHDCI Digital Data and Control connector on the PXIe-6571, use a mating connector for the VHDCI cable from NI.

**Related information:**

- ni.com/info

## Installing the CB-2162 on the PXIe-6571

The CB-2162 (NI part number 778592-01) is a connector block and prototyping board for the digital pattern instrument.

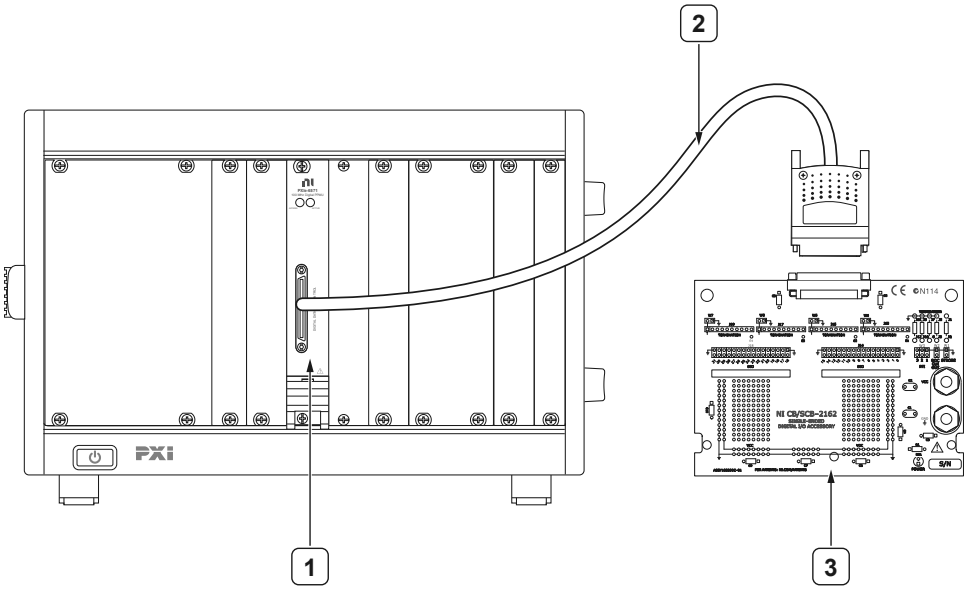
Use the CB-2162 to complete the following tasks:

- Terminate digital I/O (DIO) and control channels
- Connect with other devices for testing and debugging
- Develop and interface to prototype circuits
- Probe DIO and control channels

Complete the following steps to install the CB-2162 on a module and prepare signal connections.

> **Caution** Before connecting the cable, disconnect power from the module, accessory, and any other connected hardware to prevent damage to the hardware and personal injury. NI is not liable for damage resulting from improper connections.

**Figure 8.** CB-2162 Installation with a PXIe-6571

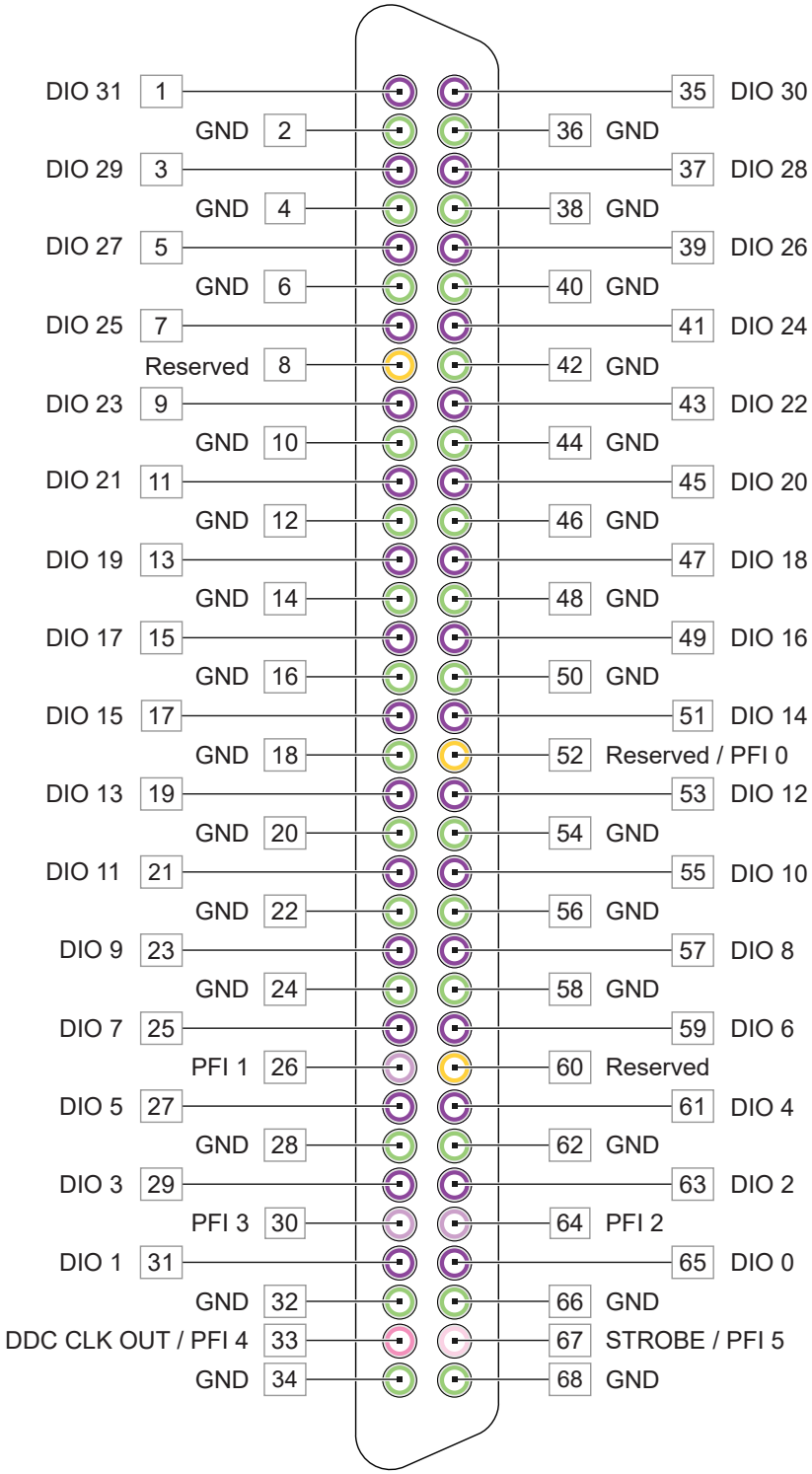



1. Chassis with the PXIe-6571 Installed
2. SHC68-C68-D4
3. CB-2162

1. Attach either end of the SHC68-C68-D4 shielded cable to the VHDCI DDC connector on the front panel of the PXIe-6571 and secure the cable with the captive screws on the cable connector.
2. Attach the other end of the cable to the DDC connector of the CB-2162 and secure the cable with the captive screws on the cable connector.
3. Make signal connections between your device channels and the CB-2162 channels.

Refer to the ***CB-2162 User Guide*** for detailed procedure.

**Figure 9.** CB-2162 DDC Connector Pinout




| DDC Pin | Signal Description |
|---|---|
| DIO <0..31> | Bidirectional digital data channels 0 through |

| DDC Pin | Signal Description |
|---|---|
| | 31. |
| STROBE/PFI_5 | External sample clock source for pattern acquisition or general-purpose PFI. |
| DDC CLK OUT/PFI_4 | Exported sample clock signal or general-purpose PFI. |
| PFI <0..3> | Programmable functional interface (PFI) channels 0 through 3 |
| GND | Ground reference for signals. |
| RESERVED | These channels are reserved for system use. Do not connect signals to these channels. |

**Caution** Connections that exceed any of the maximum ratings for the CB-2162 or the digital pattern instrument can damage the device and the computer. NI is not liable for any damages resulting from such signal connections. Refer to ***CB-2162 User Guide*** at ni.com/docs for details about the maximum input ratings.

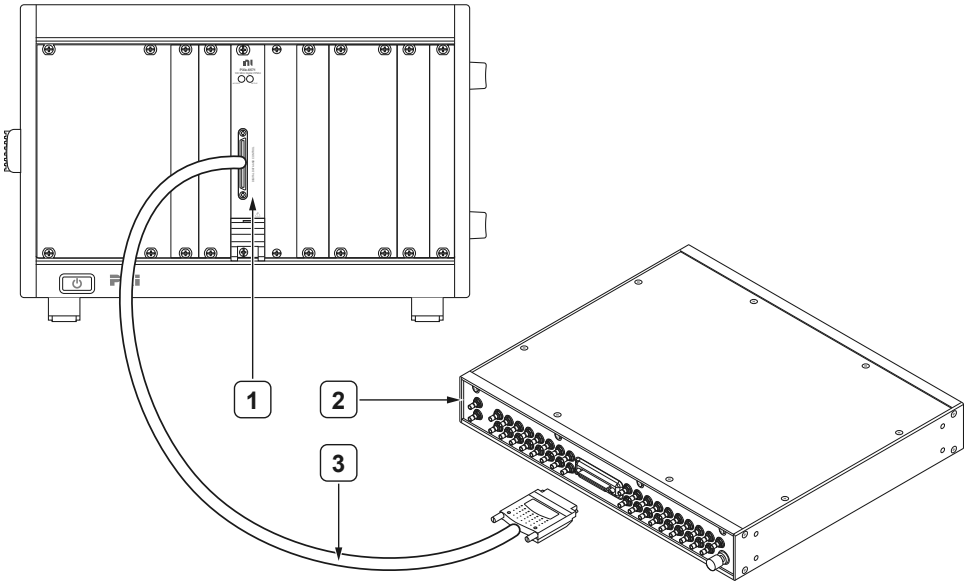
## Installing the SMB-2163 on the PXIe-6571

The SMB-2163 (NI part number 778747-01) is an SMB breakout box for high-speed digital devices. This breakout box connects to other devices for testing and debugging by routing every DIO and control channel on a separate SMB coaxial connector.

Complete the following steps to install the SMB-2163 on a module and prepare signal connections.

**Caution** Disconnect power from the device, accessory, and any other connected hardware before connecting the cable to prevent damage to the hardware and personal injury. NI is not liable for damage resulting from improper connections.

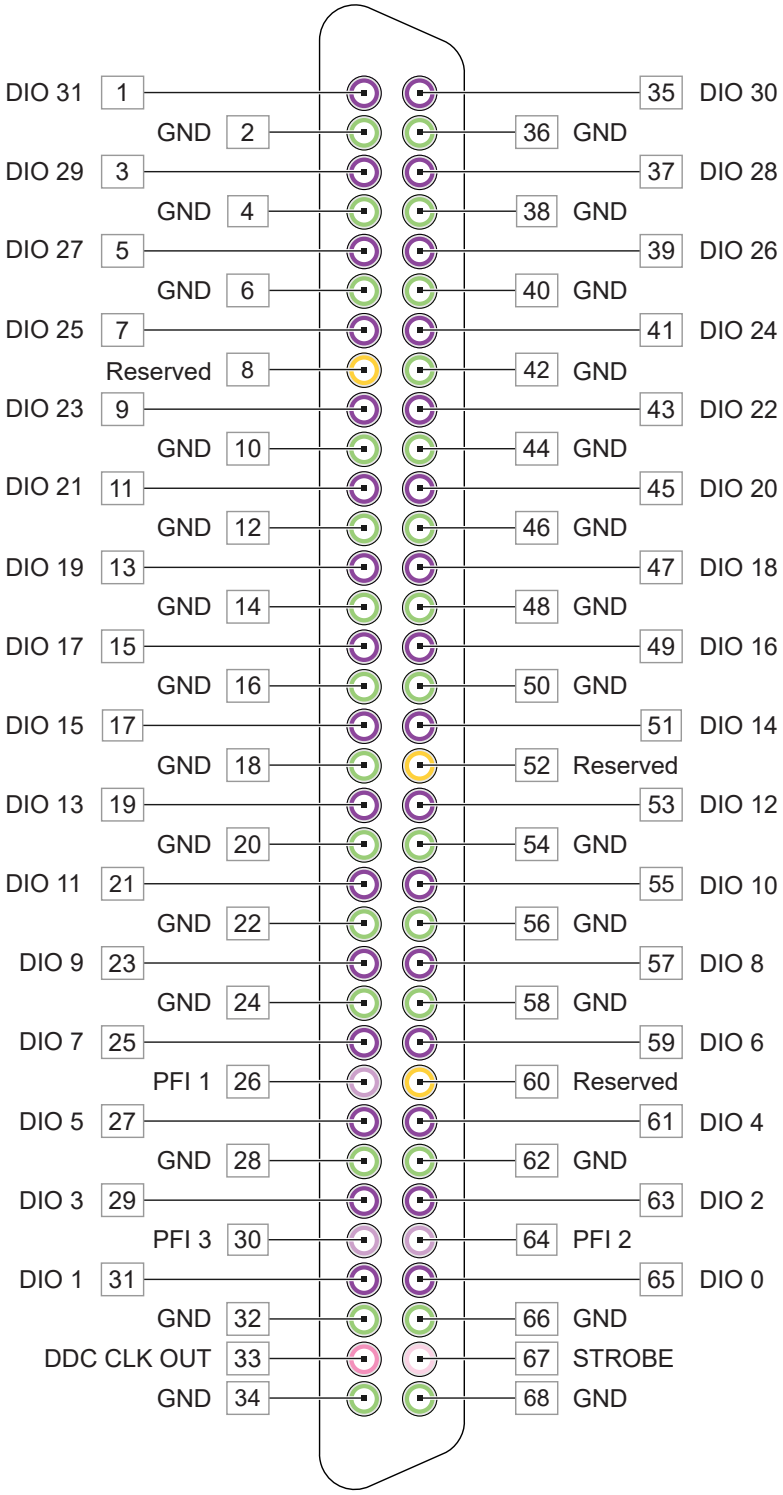
**Figure 10.** SMB-2163 Installation with a PXIe-6571




1. Chassis with the PXIe-6571 Installed
2. SMB-2163
3. SHC68-C68-D4

1. Attach either end of the SHC68-C68-D4 shielded cable to the VHDCI DDC connector of on the front panel of the PXIe-6571 and secure the cable with the captive screws on the cable connector.
2. Attach the other end of the cable to the DDC connector of the SMB-2163 and secure the cable with the captive screws on the cable connector.
3. Make signal connections by connecting SMB cables to SMB-2163 signal terminals.

**Figure 11.** SMB-2163 DDC Connector Pinout




| Item | DDC Pin | Signal Description |
|---|---|---|
| 1 | STROBE | Bidirectional digital data channels 0 through 31. |

| Item | DDC Pin | Signal Description |
|---|---|---|
| 2 | DDC CLK OUT | External Sample clock source for pattern acquisition. |
| 3 | DIO <0..31> | Exported Sample clock signal. |
| 4 | 68-Pin Digital Data & Control (DDC) Connector | Programmable function interface channels 1 through 3. |
| 5 | Programmable Function Interface (PFI) <1..3> | Ground reference for signals. |
| 6 | Shield Ground Lug | These channels are reserved for system use. Do not connect signals to these channels. |

**Caution** Connections that exceed any of the maximum ratings for the SMB-2163 or the digital pattern instrument can damage the device and the computer. NI is not liable for any damages resulting from such signal connections. Refer to ***SMB-2163 User Guide*** at ni.com/docs for details about the maximum input ratings.

**Note** To ensure a solid ground connection, tighten the SMB connectors by gently snapping them into place.

**Note** For additional shielding, you can connect the shield ground connector on the SMB-2163 to earth/hard ground. This terminal is connected to the shielded enclosure ground.

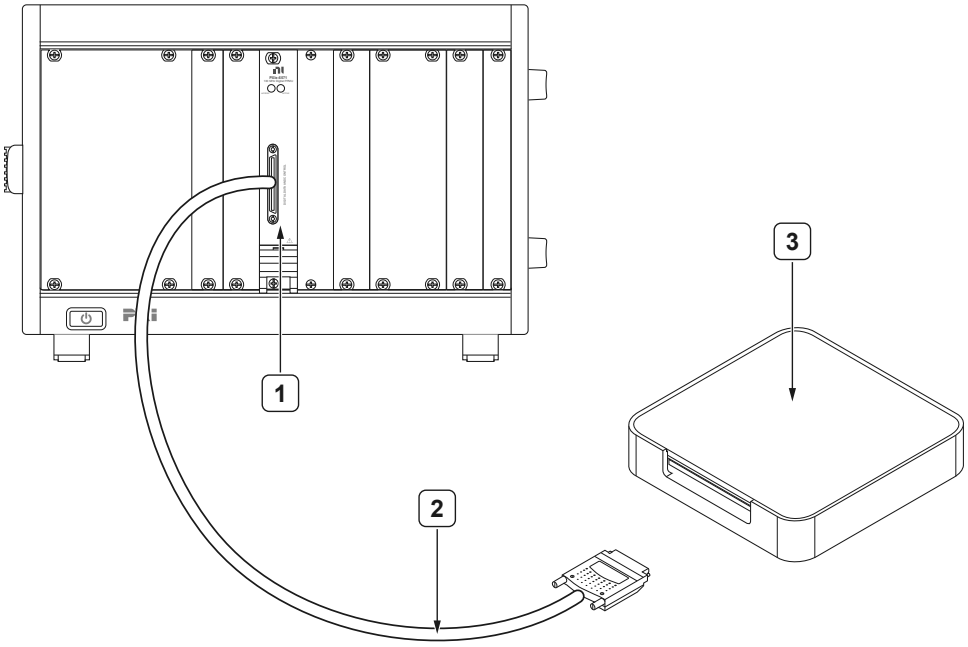
## Installing the SCB-68 HSDIO on the PXIe-6571

The SCB-68 HSDIO is a shielded I/O connector block with 72 screw terminals for connecting signals to a 68-pin very-high-density cable interconnect (VHDCI) connector.

Complete the following steps to install the SMB-2163 on a module and prepare signal connections.

**Caution** Disconnect power from the device, accessory, and any other connected hardware before connecting the cable to prevent damage to the hardware and personal injury. NI is not liable for damage resulting from improper connections.

**Figure 12.** SCB-68 HSDIO Installation with a PXIe-6571




1. Chassis with PXIe-6571 Installed
2. SHC68-C68-D4 Shielded Cable
3. SCB-68 HSDIO

1. Set up the SCB-68 HSDIO. Refer to the ***SCB-68 HSDIO Connection Guide and Specifications*** for detailed procedure.
2. Attach either end of the SHC68-C68-D4 shielded cable to the VHDCI DDC connector on the front panel of the PXIe-6571 and secure the cable with the captive screws on the cable connector.
3. Attach the other end of the cable to the DDC connector of the SCB-68 HSDIO and secure the cable with the captive screws on the cable connector.
4. Make signal connections between your device channels and the SCB-68 HSDIO channels.

**Related information:**

- SCB-68 HSDIO Connection Guide and Specifications

## Installing the SHC68-H1X38

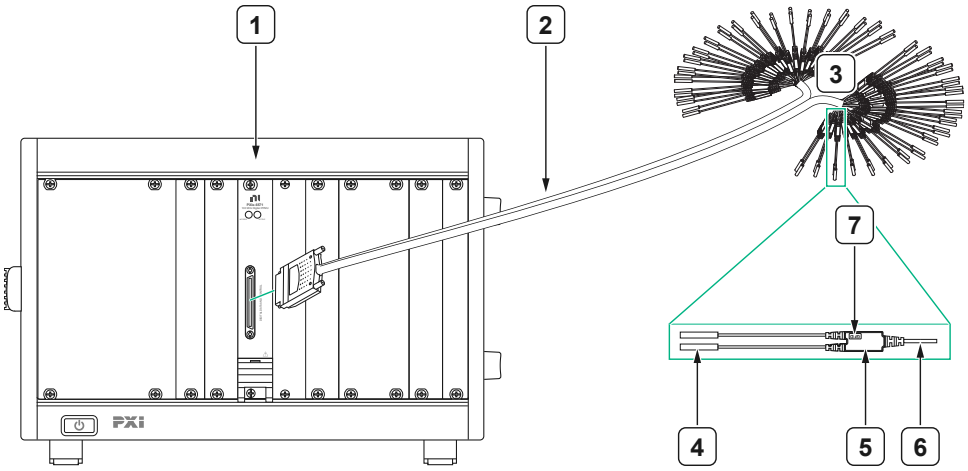
The SHC68-H1X38 cable breaks out each of 32 digital data channels through a 50 Ω characteristic impedance microcoxial cable, which is then split into two leads with receptacles.

Use this cable for acquisition and generation operations with your digital pattern instrument.

Complete the following steps to install the SHC68-H1X38 on a module and prepare signal connections.

> **Caution** Disconnect power from the device, accessory, and any other connected hardware before connecting the cable to prevent damage to the hardware and personal injury. NI is not liable for damage resulting from improper connections.

**Figure 13.** SHC68-H1X38 Parts Locator




1. Chassis with the PXIe-6571 Installed
2. SHC68-H1X38 Cable
3. Lead Pairs
4. 0.25 in. Diameter 0.1 in. Pitch Compatible Receptacles
5. Strain-Relief and Label
6. Micro-Coaxial Wire

7. Ground

1. Attach the DDC connector of the SHC68-H1X38 cable to the VHDCI DDC connector on the front panel of the PXIe-6571 and secure the cable with the captive screws on the cable connector.
2. Connect both the signal and ground receptacles and twist the pair of leads. Refer to ***SHC68-H1X38 Getting Started Guide*** for detailed information.

| Channel | Signal Description | Bundle |
|---|---|---|
| DIO <0..31> | Bidirectional digital data channels 0 through 31. | ◦ DIO<0..15>: Bundle A<br>◦ DIO<16..31>: Bundle B |
| STROBE/PFI 5 | For devices that have a STROBE channel, this signal is the external Sample clock source for dynamic acquisition. Otherwise, this is programmable functional interface (PFI) 5. | Bundle A |
| CLKOUT/PFI 4 | For devices that have a DDC CLK OUT channel, this signal is the exported sample clock signal. Otherwise, this is PFI 4. | Bundle A |
| PFI <1..3> | Programmable functional interface PFI <1..3>. | ◦ PFI 1: Bundle A<br>◦ PFI <2..3>: Bundle B |
| GND/PFI 0 | For devices that have PFI 0 on a front-panel SMB connector, this channel is ground and should not be used. Otherwise, this channel serves as PFI 0. | Bundle B |
| GND | Each channel is split into a pair of leads. The lead attached with the black wire is the GND signal, which serves as the ground reference for that particular channel. | N/A |

**Related information:**

- SHC68-H1X38 Getting Started Guide

# Triggers and Events

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

## Triggers

A ***trigger*** is an input signal received by the digital pattern instrument that causes the instrument to perform an action, such as starting a pattern burst or breaking out of a loop in a pattern.

Triggers can be internal (software-generated) or external. You can export external triggers and use them with events to synchronize hardware operation with external circuitry or other instruments.

## Events

An ***event*** is a signal generated by the digital pattern instrument that indicates when a hardware condition has been met.

You typically configure events for a specific hardware condition and then export those events for use in the test program or export them to a PXI trigger line to cause an action in another instrument configured to wait for a trigger on the same PXI trigger line. You can route up to four events to any of the eight PXI chassis backplane trigger lines per digital pattern instrument based on the signal opcodes you use in a pattern.

# Supported Triggers and Events

## Start Trigger

The Start Trigger starts execution of a pattern burst. Using a Start Trigger, synchronized digital pattern instruments configured to execute the same pattern can execute each cycle at the same time.

**Note** Some pattern functionality, such as `matched` and `failed` opcode parameters, requires use of the PXIe-6674T. Refer to the ***Synchronizing Multiple Instruments*** section for more information.

The supported trigger types for the Start Trigger are Software and Digital Edge.

- **Software Start Trigger**—This Trigger is generated by a programmatic call, such as a LabVIEW VI, .NET method, or C function, and can occur at any time based on the conditions the test program specifies.
- **Digital Edge Start Trigger**—This Trigger can be used to import external triggers from any of the eight backplane PXI chassis trigger lines. The Digital Edge Start Trigger uses external triggers to synchronize hardware operation with external circuitry or other instruments.

## Conditional Jump Trigger

The Conditional Jump Trigger controls conditional flow control opcodes, such as jump_if, in a digital pattern. They can be asserted by a digital edge, such as one from an external instrument, or by software.

The supported trigger types for the Conditional Jump Trigger are Software and Digital Edge.

## History RAM Trigger

History RAM acquires cycle information during a pattern burst. Before bursting a pattern, History RAM can be configured to trigger on First Failure, Pattern Label, or Cycle Number.

- **First Failure**—Specifies starting acquiring pattern information in History RAM on the first failed cycle in a pattern burst.
- **Pattern Label**—Specifies starting acquiring pattern information in History RAM starting from a specified cycle number.
- **Cycle Number**—Specifies starting acquiring pattern information in History RAM starting from a specified pattern label, augmented by vector and cycle offsets.

## Pattern Opcode Event

A Pattern Opcode Event is a signal that the digital pattern instrument generates under

opcode control. You can export the Pattern Opcode Event to any of the eight backplane PXI chassis trigger lines. You typically configure events to indicate when a specific hardware condition has been met and then export events for use in the test program. You can configure up to four Pattern Opcode Events. The `set_signal`, `clear_signal`, and `pulse_signal` opcodes in the pattern control the Pattern Opcode Events.

**Related reference:**

- Synchronizing Multiple Instruments

## Signal Routing

The possible signal routes depend on the digital pattern instrument, the PXI Express chassis, and the occupied PXI Express chassis slot. You can use the PXI Express trigger bus to send and receive events and triggers. You can characterize signal routing operations by source and destination. The following table summarizes the possible sources and destinations for PXIe-6571 signals.

**Table 6.** Trigger Routing for PXIe-6571

| **Source** | | **Destination** | | |
|---|---|---|---|---|
| | | **Backplane Destination** | **Internal Destination** | |
| | | **PXI_TRIG <0..7>** | **Start Trigger** | **Conditional Jump Trigger <0..3>** |
| Backplane Source | PXI_TRIG <0..7> | — | ✓ | ✓ |
| Internal Source | Start Trigger | ✓ | — | — |
| | Software Trigger | — | ✓ | ✓ |
| | Pattern Opcode Event <0..3> | ✓ | — | — |
| | Conditional Jump Trigger <0..3> | ✓ | — | — |

## Creating Static Signal Routes

If your PXI Express chassis consists of multiple trigger bus segments, you may need to route signals across them depending on the slot location of your digital pattern instruments. The NI-Digital Pattern Driver API creates a dynamic signal route automatically when you provide the fully qualified name for the source and destination signals.

If you are using a third-party chassis that does not support the PXI Trigger Management specifications or the instrument that you are routing signals to does not support dynamic routing, you may need to create a static route by manually configuring the chassis. Complete the following steps to configure a static route:

**Note** No additional configuration is required to use dynamic signal routes.

1. Launch MAX by navigating to **Start** » **All Programs** » **NI MAX** or by double-clicking the NI MAX desktop icon.
2. Expand **Devices and Interfaces** and then expand the **Chassis** tree.
3. Select your chassis.
   The attributes of your chassis are displayed on the right side of the MAX window.
4. Click the **Triggers** tab below the attributes view.
   A table in the **Triggers** view shows the PXI trigger bus segments of your chassis.
5. Select the checkbox for the PXI trigger line to be used and the bus segment that the signal source is in.
6. In the drop-down menu to the right of the PXI trigger line, make the selection to route away from the sourcing bus segment.

   **Note** Using a PXI trigger line in a static route limits the number of dynamic routes you can create.
7. Save the changes.

To release a static route and make it available for dynamic routing, complete the following steps:

1. Launch MAX.
2. Deselect the checkbox for the reserved PXI trigger line.

3. In the drop-down menu to the right of the PXI trigger line, select **Dynamic**.

## Synchronizing Multiple Instruments

You can synchronize digital pattern instruments to varying degrees.

- You can operate instruments independently with triggers and events passed between them to coordinate operations.
- You can use the NI-TClk API to phase align the internal clocks of all the instruments and route triggers deterministically so that all the instruments respond to and receive triggers at the same time.
- In the Digital Pattern Editor, you can use the same instrument group name in the pin map for multiple instruments, which automatically phase aligns the internal clocks of the instruments included in the group and routes triggers deterministically.



**Note** You can group digital pattern instruments of only the same model and channel count, and the instruments must be in the same chassis. For example, you can group two PXIe-6571s of matching channel count that are within the same chassis but not a PXIe-6571 and PXIe-6570 in the same chassis.



**Note** You can open a synchronized multi-instrument session by passing multiple instruments into the Initialize with Options API. Refer to the ***Digital Pattern User Manual*** for more information.

After aligning the internal clocks and routing triggers, you can use an additional module in the chassis, a PXIe-6674T, to combine pattern comparison results across digital pattern instruments and control subsequent pattern execution based on those results. Refer to the ***PXIe-6674T User Manual*** for information about installing PXIe-6674T in the timing and synchronization slot of the chassis.



**Note** You cannot disable the synchronization without recreating the instrument session.

## Internal Clock Phase Alignment

For digital pattern instruments, the MasterClk internal clock is locked to the PXIe_CLK100 signal. To ensure that other instruments synchronize with a digital pattern instrument, their internal clocks, which are sometimes called Sample clocks, must also be locked to PXIe_CLK100.

## System Timing Alignment

The NI-TClk API included with an installation of the NI-Digital Pattern Driver aligns the timing reference on synchronized instruments. For instruments used inside a Semiconductor Test System (STS), the timing calibration procedure aligns the timing across all digital instrument channels. On systems without STS timing calibration, enable the Timing Absolute Delay property to adjust the timing reference on synchronized instruments for better timing alignment, measured in seconds.

## Trigger Routing

NI-TClk uses the PXI trigger bus to deterministically route triggers among multiple instruments to ensure that all instruments in the system execute at the same time. Deterministic trigger routing ensures that all instruments in the system start at the same time. Refer to multi-device NI-TClk examples for more information about how to use the NI-TClk API and refer to the ***NI-TClk Synchronization Help*** for more information on using NI-TClk to synchronize instruments.

## Using Matched and Failed Opcode Parameters with Synchronized Digital Pattern Instruments

During a pattern burst, the digital pattern instrument pattern sequencer controls the `matched` and `failed` parameters of the `jump_if` and `exit_loop_if` opcodes. When you burst patterns that use the `matched` and `failed` parameters across multiple synchronized digital pattern instruments, you must complete the following steps to configure the system to combine comparison results across the digital pattern instruments.

- Use the NI-Sync 16.1 or later to initialize the PXIe-6674T timing and synchronization instrument.

- Use the niTClk Synchronize API to align the internal clocks and route the triggers of the digital pattern instruments.
- Use the niDigital Enable Match Fail Combination API to enable each synchronized instrument to process the combined comparison results.

When you configure a system to combine matched and failed results, if a failed condition occurs on any of the sites enabled for bursting on any of the multiple synchronized digital pattern instruments, the combined system result for the failed condition is TRUE and influences the flow of pattern execution through opcodes that use the failed condition as a parameter. Calling the niDigital Get Site Pass Fail API in the test program on all digital pattern instruments returns failed results only for the sites on the instruments that caused the failure.

When you configure a system to combine matched and failed results, the combined system result for the matched condition is TRUE only when the matched condition is satisfied by all sites enabled for bursting across multiple synchronized digital pattern instruments that include pins in the specified match condition.

**Table 7.** Common Use Cases and Behavior

| Use Case | Behavior |
|---|---|
| • Single PXI Express chassis<br>• Single digital pattern instrument<br>• Multiple sites<br>• Without PXIe-6674T | If you require per-site matched and failed capability, create a site loop in the test program to burst the pattern for each site sequentially. |
| • Single PXI Express chassis<br>• Multiple digital pattern instruments<br>• Multiple sites<br>• Without PXIe-6674T | • The pattern sequencer in each digital pattern instrument might not execute the same vector at the same time.<br>• If you assign a single site per digital pattern instrument, each site can process the matched and failed parameters independently and independently execute conditional opcodes in parallel.<br>• If you assign multiple sites per digital pattern instrument and require per-site matched and failed capability, create a site |

| Use Case | Behavior |
| --- | --- |
| | loop in the test program to burst the pattern for each site sequentially. |
| • Single PXI Express chassis<br>• Multiple digital pattern instruments<br>• Multiple sites<br>• With PXIe-6674T | • All digital pattern instruments within a single chassis become a single digital subsystem and execute the same vector at the same time, including flow control operations. Instruments and sites assigned to this digital subsystem do not branch independently.<br>• If you require per-site matched and failed capability, create a site loop in the test program to burst the pattern for each site sequentially. |
| Multiple PXI Express chassis | Each chassis operates independently. |

### Related reference:

- Supported Triggers and Events
- Clocking

### Related information:

- PXIe-6674T User Manual

# Making Frequency Measurements

You can measure the signal frequency on specified channels by using the following:

- niDigital Frequency Counter Configure Measurement Time VI
- niDigital Frequency Counter Measure Frequency VI
- DigitalFrequencyCounter.Configure .NET property
- DigitalFrequencyCounter.MeasureFrequency .NET method
- niDigital_FrequencyCounter_ConfigureMeasurementTime C function
- niDigital_FrequencyCounter_MeasureFrequency C function

Use the $V_{OL}$ comparators to count the number of rising edges through the $V_{OL}$ threshold over the measurement time. To ensure accurate measurements, ensure that the input signal produces at least five rising edges within the measurement time, is free-running, and has monotonic rising and falling-edges through the $V_{OL}$ threshold.

The PXIe-6571 has a discrete frequency counter per eight digital input channels to measure the frequency on those channels. For the PXIe-6571 (32-channel), you can optimize frequency measurements for test time if the channels you want to measure are from different frequency counters.

**Table 8.** Frequency Counters and Associated Channels

| Frequency Counter | Channels |
|---|---|
| 1 | <0..7> |
| 2 | <8..15> |
| 3 | <16..23> |
| 4 | <24..31> |

Refer to ***PXIe-6571 Specifications*** for the supported frequency measurement ranges.

**Related information:**

- PXIe-6571 Specifications

# Correctly Measuring Signal Integrity

For applications requiring the highest levels of signal integrity, visit ni.com/info and enter the Info Code `MeasureHSDigital` to access ***Measuring Digital Signal Integrity with an Oscilloscope*** for more information.

# Sourcing and Capturing Waveforms

Source and capture functionality allows data to be inserted into or extracted from a pattern burst. Use source or capture functionality when the data you need to use is site-specific or only determined at run time. For example, if there is a digital protocol associated with passing data to or from the DUT, the pattern defines the protocol, such as clocking or chip selects, but you can source the data from source memory or capture the data into capture memory.

The source and capture functionality is useful in the following cases:

- When the data you need to use can be separate from the pattern, such as tests that require site-specific data, data stimulus to a DAC, or data response from an ADC.
- If the data that needs to be written is only determined at run time, such as register reads or writes.

The functionality prevents the need to compile new patterns in the middle of a test execution or to have all possible response patterns pre-loaded. Each digital pattern instrument includes source and capture functionality for up to eight sites. You can send source waveforms and receive capture waveforms through any of the pins on the digital pattern instrument. Refer to the ***PXIe-6571 Specifications*** for more information about the number of channels you can connect for each digital pattern instrument.

**Note** Ensure that the total average data rate for the source or capture data falls within the maximum source or capture rate for the digital pattern instrument. Visit ni.com/info and enter the Info Code `DigitalSourceCapture` to access ***Calculating Source and Capture Bandwidth for Digital Pattern Instruments*** for more information.

**Note** To use source and capture functionality that spans multiple digital pattern instruments synchronized with NI-TClk, you must configure the functionality on each synchronized digital pattern instrument.

Refer to the ***Digital Pattern User Manual*** for more information about NI-Digital Pattern Driver source and capture API, source and capture configuration in Digital Pattern Editor, and source and capture opcodes.

**Related information:**

- PXIe-6571 Specifications

# Sourcing Waveforms

## Configuring Source Waveforms

Use the source waveform configuration document in the Digital Pattern Editor or the NI-Digital Pattern Driver source API to configure source waveforms.

You can configure source waveforms for serial or parallel use:

- In ***serial*** waveform type, each vector that contains a `source` opcode serially shifts 1 bit on the pin using the drive from source pin state (D) to indicate what value to drive. The data width of the serial source sample is set up at configuration, and after an appropriate number of bits have been shifted, the source engine moves on to the next sample.
- In ***parallel*** waveform type, the entire waveform sample is sourced on all specified pins in parallel for each vector that contains a `source` opcode. Use the drive from source pin state (D) for each pin on each such vector that corresponds to the pins to source.

**Note** Ensure that the total average data rate for the source or capture data falls within the maximum source or capture rate for the digital pattern instrument. Visit ni.com/info and enter the Info Code `DigitalSourceCapture` to access ***Calculating Source and Capture Bandwidth for Digital Pattern Instruments*** for more information.

When configuring serial and parallel source waveforms, you can specify one of the following data mapping options:

- **Site Unique**—Sources unique waveform data to each site.
- **Broadcast**—Writes the same waveform data to all sites if multiple sites are configured in the pin map.

## Sending Source Waveforms

You can send source waveforms at run time using opcodes in the pattern file. Use the `source_start` and `source` opcodes to send waveform data when you burst the pattern. Use the `source_d_replace` opcode to change the behavior of the source data.

## Source Considerations

- You can send multiple source waveforms with the same configuration in a single pattern burst.
- You can load up to 512 source waveforms on the instrument at a time.
- The total number of pins supported by source functionality is 32.

Refer to the ***PXIe-6571 Specifications*** for more information about source capacity.

**Related information:**

- PXIe-6571 Specifications

# Capturing Waveforms

## Configuring Waveforms

You can configure capture functionality for serial or parallel use.

Use the capture waveform configuration document in the Digital Pattern Editor or the NI-Digital Pattern Driver capture API to configure capture waveforms.

You can configure capture waveforms for serial or parallel use:

- In ***serial*** waveform type, each vector that contains a `capture` opcode serially shifts 1 bit on pins using the pin state V. The data width of the serial capture sample is set up at configuration, and after an appropriate number of bits have

been shifted, the capture engine moves on to the next sample.

- In ***parallel*** waveform type, the entire waveform sample is captured on all specified pins in parallel for each vector that contains a `capture` opcode. Use the pin state V for each pin on each such vector that corresponds to the pins for which you want to capture waveform samples.

**Note** Capture memory stores a single bit per channel per capture cycle. A 1 is stored for any line where $V_{out} > V_{oh}$ and $V_{out} > V_{ol}$. Otherwise, a 0 is stored. M pin states are seen as 0 in capture memory.

**Note** Ensure that the total average data rate for the source or capture data falls within the maximum source or capture rate for the digital pattern instrument. Visit ni.com/info and enter the Info Code `DigitalSourceCapture` to access ***Calculating Source and Capture Bandwidth for Digital Pattern Instruments*** for more information.

## Receiving Capture Waveforms

You can receive capture waveform data at run time using opcodes in the pattern file. Use the `capture_start`, `capture`, and `capture_stop` opcodes to store values when you burst a pattern.

## Capture Considerations

- You can receive multiple capture waveforms with the same configuration in a single pattern burst.
- You can use up to 512 capture waveforms on the digital pattern instrument at a time.
- The total number of pins supported by capture functionality is 32.

Refer to ***PXIe-6571 Specifications*** for more information about the number of samples you can capture for all patterns in a burst operation.

**Related information:**

- PXIe-6571 Specifications

# Scan Testing

Scan is a technique for testing DUTs designed with internal test circuitry to validate that the digital logic in the DUT was properly manufactured. You can create scan patterns to test DUT circuitry using scan chains. Scan patterns contain scan pins and scan vectors in addition to DUT pins and vectors. By serially shifting data into and out of the scan test pins, you can configure the DUT to test its functional circuitry. Dedicated scan features of the digital pattern instrument can help create patterns for scan testing.

To use the scan patterns, you can complete the following:

- Create scan patterns as text or text pattern files (.digipatsrc) and then import the patterns into a Digital Pattern Editor project.
- Create a regular pattern in the Digital Pattern Editor and then add scan pins and scan vectors to the pattern.

## Scan Pins and Chains

You can select pattern pins for scan input or scan output. Scan input pins are restricted to pin states 0 and 1, and scan output pins are restricted to pin states L, H, and X. These restrictions apply only when you use these pins with a scan opcode. On non-scan vectors, these pins are regular DUT pins.

A scan chain is a pair of scan input pins and scan output pins.

## Scan Vectors

***Scan vectors*** are pattern vectors with the scan opcode. When the digital pattern instrument executes a scan vector, scan pins execute the scan pin data defined in the pattern. Non-scan pins repeats the pin state defined on the scan vector until the scan vector finishes.

You can call the Fetch History RAM Scan Cycle Numbers API to get the scan cycle

numbers from the scan vector. Cycles that are not part of a scan vector return a -1 for the scan cycle numbers.

Scan vectors are compressed in vector memory. Using fewer scan chains per digital pattern instrument results in better vector memory compression. Splitting scan chains across digital pattern instruments can improve vector memory compression but could negatively impact performance of capture memory or History RAM when scan pins are used outside of scan vectors.

# Example Programs

NI-Digital Pattern Driver includes several examples that demonstrate how to use NI-Digital Pattern Driver to configure and control digital pattern instruments.

## 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 "Digital Pattern" to locate all NI Digital Pattern 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 or LabVIEW NXG, select **Help** » **Find Examples** and navigate to **Hardware Input and Output** » **Modular Instruments** » **NI-Digital Pattern Driver**.

## Example Programs

To locate programming and operating examples, refer to the following table.

**Table 9.** Installed NI-Digital Pattern Driver Example Locations

| Option | Installed Example Location |
|---|---|
| LabVIEW | `<Program Files>\National Instruments\<LabVIEW>\examples\instr\niDigital`, where <LabVIEW> is the directory for the specific LabVIEW version that is installed. |
| C | `<Public Documents>\National Instruments\NI-Digital\Examples\C` |
| .NET | `<Public Documents>\National Instruments\NI-Digital\Examples\DotNET 4.x` |

**Note** To access Microsoft .NET and C examples, you can navigate to **National Instruments » NI Digital Pattern Examples** from the Windows Start menu.

## Getting Started with Digital Pattern Instruments

Use the getting started examples, located by default in the `<Public Documents>\National Instruments\NI-Digital\Examples\Getting Started` directory, help you learn key concepts. You can also access the getting started examples by selecting the **Start** menu and then navigating to NI Digital Pattern Examples in the National Instruments folder.

Additionally, the Digital Pattern Editor includes examples for getting started. In the Welcome window, click **Examples** or click the **Learning** tab to launch the Learning window, which includes examples of common usages of digital patterns.

Use the NI-Digital Pattern Driver examples to learn more about how to use the NI-Digital Pattern Driver software to configure and control digital pattern instruments. Use the NI Example Finder to locate LabVIEW and LabVIEW NXG examples. The Microsoft .NET examples are located in the `<Public Documents>\National Instruments\NI-Digital\Examples\DotNET 4.x` directory. The C examples are located in the `<Public Documents>\National Instruments\NI-Digital\Examples\C` directory.

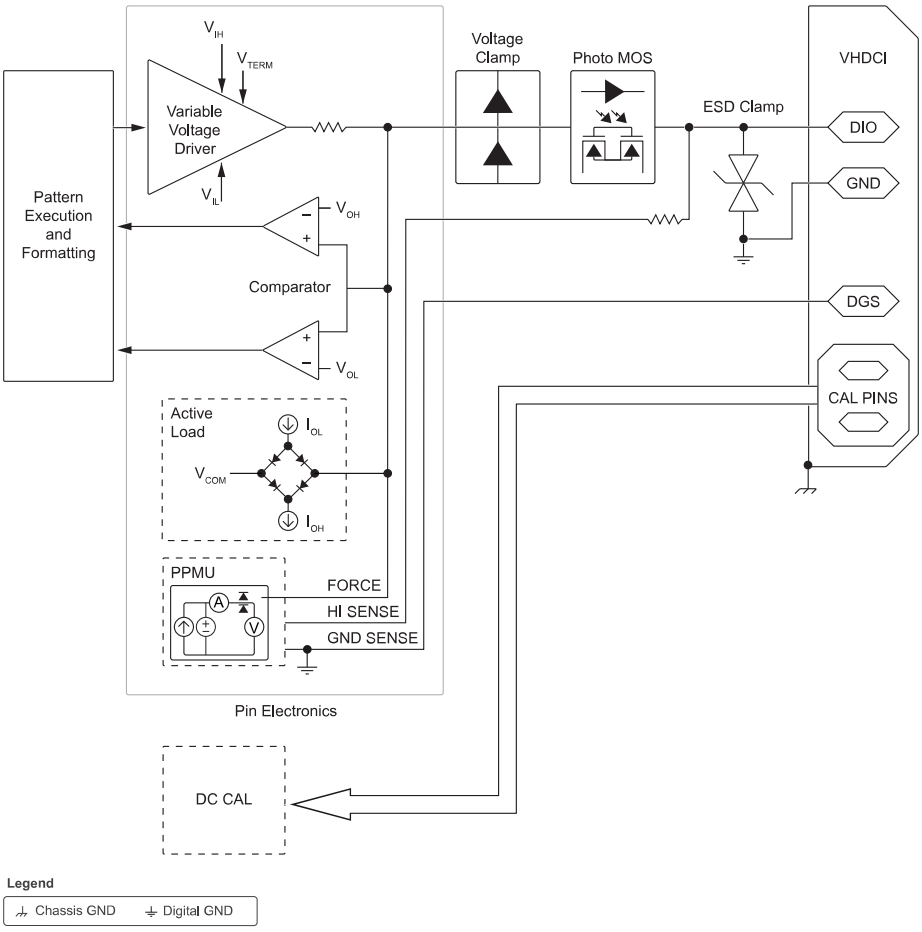
# PXIe-6571 Operating Guidelines

Refer to the following sections for information about PXIe-6571 features and guidelines for operating the PXIe-6571.

## Pin Electronics

Pin electronics provide the electrical interface to the DUT and allow the engineer to drive or receive digital data and emulate the conditions of other loads and components interacting with the device. Each channel on the PXIe-6571 has a pin driver, comparator, active load, and PPMU. You need to specify levels from the perspective of the DUT, not the digital pattern instrument.

**Figure 14.** PXIe-6571 Pin Electronics Block Diagram




# Programming Pin Electronics

## Selecting Pin Function

The digital pattern instrument provides multiple capabilities in one instrument. Use the niDigital Select Function VI, the SelectedFunction .NET enumeration, or the niDigital_SelectFunction C function to programmatically set the pin function to one of the following options.

- ***Digital***
- ***PPMU***
- ***Off***
- ***Disconnect***

| Selected Function | Description | Usage |
|---|---|---|
| Digital | The pin is connected to the driver, comparator, and active load functions. The PPMU is not sourcing but can make voltage measurements. When you change the selected function to Digital, the state of the digital pin driver returns to the last digital state on the line.<br><br>**Note** You must be in Digital mode to use the frequency counter. | To control the state of the digital pin driver through software, use the following:<br><br>• niDigital Write Static VI<br>• DigitalPinSet.WriteStatic .NET method<br>• niDigital_WriteStatic C function<br><br>To control the state of the digital pin driver through a pattern, use the following:<br><br>• niDigital Burst Pattern VI<br>• DigitalPatternControl.BurstPattern .NET method<br>• niDigital_BurstPattern C function<br><br>To automatically switch the selected function of the pins in the pattern burst to Digital, set the `select digital function` parameter of following to TRUE:<br><br>• niDigital Burst Pattern VI<br>• DigitalPatternControl.BurstPattern .NET method<br>• niDigital_BurstPattern C function |
| PPMU | The pin is connected to the PPMU. The driver, comparator, and active load are off while this function is selected. | To source a voltage or current, call the following:<br><br>• niDigital PPMU Source VI<br>• DigitalPpmu.Source .NET method<br>• niDigital_PPMU_Source C function<br><br>The niDigital PPMU Source VI, the DigitalPpmu.Source .NET method, and the niDigital_PPMU_Source C function |

| Selected Function | Description | Usage |
|---|---|---|
| | | automatically switch the selected function to the PPMU state and start sourcing from the PPMU. Changing the selected function to Disconnect, Off, or Digital causes the PPMU to stop sourcing.<br><br>If you change the selected function to PPMU using the niDigital Select Function VI, the DigitalPinSet.SelectedFunction .NET property, or the niDigital_SelectFunction C function, the PPMU is initially not sourcing.<br><br>**Note** You can make PPMU voltage measurements using the niDigital PPMU Measure VI, the DigitalPpmu.Measure .NET method, or the niDigital_PPMU_Measure C function from within any selected function. |
| Off | The pin is electrically connected, and the PPMU and digital pin driver are off while this function is selected. | — |
| Disconnect | The pin is electrically disconnected from instrument functions. Selecting this function causes the PPMU to stop sourcing prior to disconnecting the pin. | — |

| Selected Function | Description | Usage |
|---|---|---|
| | **Notice** In the disconnect state, some I/O protection and sensing circuitry remains exposed. Do not subject the instrument to voltages beyond its operating range. | |

### Defining Levels

Define levels for digital pins or digital pin groups in the levels file (`.digilevels`) with the Digital Pattern Editor or through a test program written in LabVIEW, C, or with Microsoft .NET technologies.

You can program levels on a per-pin or per-pin group basis. The levels apply to the pin or pin group for all currently enabled sites. You can also program levels on a per-channel basis. In this case, the level applies only to the specified single channel.

Refer to ***PXIe-6571 Specifications*** for valid level values.

### Multisite Programming

The digital pattern instrument includes access to the following hardware resources for up to eight sites:

- Source engine
- Capture engine
- History RAM
- PPMU

### Related information:

- PXIe-6571 Specifications

## Programming Pin Driver

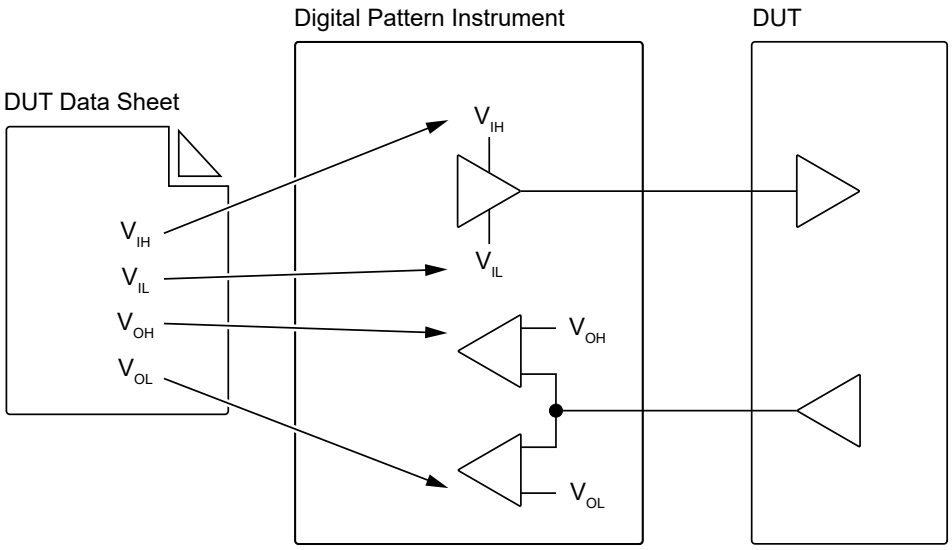
### Programmable Levels

You can program the pin driver with the following pin levels using the NI-Digital Pattern Driver or set pin levels interactively in Digital Pattern Editor. Pin levels are programmable per pin or by pin group.

- $V_{IH}$—Specifies the voltage that the instrument applies to the input of the DUT when the instrument drives a logic high (1).
- $V_{IL}$—Specifies the voltage that the instrument applies to the input of the DUT when the instrument drives a logic low (0).
- $V_{TERM}$—Specifies the termination voltage the instrument applies during non-drive cycles when the termination mode is set to $V_{TERM}$. The instrument applies the termination voltage through a 50 Ω parallel termination resistance.

The NI-Digital Pattern Driver does not enforce a minimum separation between $V_{IH}$ and $V_{IL}$. Refer to ***PXIe-6571 Specifications*** for information about the minimum separation required for reliable results.

The following diagram illustrates pin levels as they relate to the digital pattern instrument and the DUT.




### Initial State

On power-up, pins are in a high-impedance state. To avoid unexpected results for DUTs that are sensitive to the initial state of a pattern burst, programmatically specify the initial pin state by using the niDigital Write Static VI, the DigitalPinSet.WriteStatic .NET method, or the niDigital_WriteStatic C function.

You can interactively set the initial state for a pin by selecting the corresponding driver symbol in the Drive section of the Pin View pane of the Digital Pattern Editor. After each pattern burst, the ending state of the last executed vector persists until the start of the next pattern burst or until you programmatically or interactively change the pin state or pin function.

### Termination Mode

The termination mode specifies the behavior of the pin when the pin function is set to Digital and the current pin state is not a drive state. Non-drive states include L, H, M, V, X and potentially -. Pattern data and the termination mode determine the full behavior of the pin. You can set the following termination modes interactively in the Digital Pattern Editor or programmatically with the NI-Digital Pattern Driver:

- **High Z**—Specifies that, for non-drive pin states (L, H, X, V, M, E), the pin driver is put in a high-impedance state and the active load is disabled.
- **Active Load**—Specifies that, for non-drive pin states (L, H, X, V, M, E), the active

load is connected and the instrument sources or sinks a defined amount of current to load the DUT.
  - **$I_{OL}$**—Specifies the amount of current sourced by the instrument and therefore sunk by the DUT.
  - **$I_{OH}$**—Specifies the amount of current sunk by the instrument and therefore sourced by the DUT.
  - **$V_{COM}$**—Specifies the voltage at which the instrument changes between sourcing and sinking.
- **$V_{TERM}$**—Specifies that, for non-drive pin states (L, H, X, V, M, E), the pin driver terminates the pin to the configured $V_{TERM}$ voltage through a 50 Ω impedance. $V_{TERM}$ is adjustable to allow for the pin to terminate at a set level. This is useful for devices that might operate incorrectly if an instrument pin is unterminated and is allowed to float to any voltage level within the instrument voltage range. To address this issue, enable $V_{TERM}$ by configuring the $V_{TERM}$ pin level to the desired voltage and selecting the $V_{TERM}$ termination mode. Setting $V_{TERM}$ to 0 V and selecting the $V_{TERM}$ termination mode has the effect of connecting a 50 Ω termination to ground, which provides an effective 50 Ω impedance for the pin. This can be useful for improving signal integrity of certain DUTs by reducing reflections while the DUT drives the pin.

For more information about terminating digital pattern instruments, visit ni.com/info and enter the Info Code `InstrumentTermination` to access ***Terminating Digital Pattern Instruments***.

**Related information:**

- PXIe-6571 Specifications

## Programming Comparator

### Programmable Levels

You can program the comparator with the following pin levels using the NI-Digital Pattern Driver or set pin levels interactively in the Digital Pattern Editor. Pin levels are programmable per pin or by pin group.

- **$V_{OL}$**—Specifies the DUT output voltage below which the comparator on the instrument pin interprets a logic low (L).
- **$V_{OH}$**—Specifies the DUT output voltage above which the comparator on the instrument pin interprets a logic high (H).

The NI-Digital Pattern Driver does not enforce a minimum separation between $V_{OH}$ and $V_{OL}$. Refer to ***PXIe-6571 Specifications*** for information about the minimum separation required for accurate results.

**Data States Comparison**

For each pin in the current vector, the comparator evaluates the output voltage of the DUT ($V_{OUT}$) against the currently specified $V_{OH}$ and $V_{OL}$ levels at the compare strobe time. This evaluation is performed for each pin that specifies one of the following data states:

- L—Compare low. Expect $V_{OUT} < V_{OL}$.
- H—Compare high. Expect $V_{OUT} > V_{OH}$.
- V—Compare valid. Expect $V_{OUT} > V_{OH}$ or $V_{OUT} < V_{OL}$.
- M—Compare midband. Expect $V_{OL} < V_{OUT} < V_{OH}$.

If the evaluation of any pin does not match its expected value, the comparator generates a failure for that cycle. Use the `match` opcode to modify this behavior and does not generate a failure for that cycle.

**Related information:**

- PXIe-6571 Specifications

## Programming Active Load

**Programmable Levels**

The pin state for the current cycle dynamically controls the active load. You can program the active load with the following pin levels using the NI-Digital Pattern Driver or set pin levels interactively in the Digital Pattern Editor. These levels apply only when the termination mode is set to active load, the specified pins are in a non-drive pin

state (L, H, X, V, M), and the selected pin function is Digital. Pin levels are programmable per pin or by pin group.

- **$I_{OL}$**—Specifies the current that the DUT sinks from the active load while outputting a voltage below $V_{COM}$.
- **$I_{OH}$**—Specifies the current that the DUT sources to the active load while outputting a voltage above $V_{COM}$.
- **$V_{COM}$**—Specifies the commutating voltage level at which the active load circuit switches between sourcing current and sinking current.

You can use the active load to provide pull-up or pull-down functionality for an open-drain DUT, such as in an I2C interface. When the DUT drives low, the DUT sinks current sourced by the active load. When the DUT stops driving low, the active load of the digital pattern instrument continues sourcing current until the voltage on the pin reaches $V_{COM}$.

## Programming PPMU

You can use the pin parametric measurement unit (PPMU) to make DC parametric measurements on DUT pins and force voltage or current.

The PPMU supports the following operations:

- Force voltage, no measure
- Force voltage, measure voltage
- Force voltage, measure current
- Force current, no measure
- Force current, measure voltage
- Force current, measure current
- Measure voltage, no force

**Related information:**

- NI-Digital Pattern User Manual

## Force Voltage

You can perform the following operations while forcing voltage to the DUT with the PPMU:

- Force voltage, measure current
- Force voltage, measure voltage
- Force voltage, no measure

When you program or interactively set the PPMU to force voltage to the DUT, you must specify the following values associated with the voltage level. Refer to ***Digital Pattern User Manual*** for more information about PPMU DC voltage API.

- **Voltage Level ($V_F$)**—Specifies the voltage level that the PPMU forces to the DUT pin.
- **Current Limit Range (I Limit Range)**—Specifies the current range to use when forcing a voltage from the PPMU to a DUT.

  Selecting the smallest range that includes the maximum current expected to be required by the DUT provides a degree of current limiting for force voltage operations. This current limiting functionality uses a range setting rather than a current clamp, and therefore you should not rely solely on it to limit current for sensitive DUTs or test setups. If the selected current range is smaller than the actual current consumption of the DUT pin, the voltage output may not reach the specified voltage level due to the sourced current being limited.

> **Note** A voltage glitch occurs if the digital pattern instrument is already sourcing a voltage when you change the current range. For example, when switching from a smaller to a larger current range, the PPMU output impedance value decreases. If the digital pattern instrument is driving a resistor to ground, the instrument forms a voltage divider between the PPMU output impedance and the output load. As a result of the voltage divider effect, when the PPMU output impedance decreases, the output voltage instantly increases before the loop can compensate for the change. Likewise, when switching from a larger to a smaller current range, the PPMU output impedance value increases, which results in a decrease in the instantaneous voltage observed on the DIO channel.

**Note** Lower current limit ranges requires a larger settling time for the sourced voltage to reach the programmed level.

## Force Current

You can perform the following operations while forcing current to the DUT with the PPMU:

- Force current, measure voltage
- Force current, measure current
- Force current, no measure

When you program or interactively set the PPMU to force current to the DUT, you must specify the following pin levels. Refer to ***Digital Pattern User Manual*** for more information about PPMU DC current API.

- **Current Level ($I_F$)**—Specifies the current level that the PPMU forces to the DUT.
- **Current Level Range (I Level Range)**—Specifies the current range to use when forcing a current from the PPMU to a DUT. Selecting the smallest range adequate for the desired current level provides the best current accuracy for the force current and measure current operations.
- **Voltage Limit High ($V_{CH}$)**—Specifies the nominal voltage at the pin at which the high side voltage clamp activates when the PPMU forces current to the DUT.
- **Voltage Limit Low ($V_{CL}$)**—Specifies the nominal voltage at the pin at which the low side voltage clamp activates when the PPMU forces current to the DUT.

**Note** Refer to ***PXIe-6571 Specifications*** for more information about when the voltage clamps begin to conduct.

## Measure Voltage

You can perform the following voltage measurement operations with the PPMU:

- Force voltage, measure voltage
- Force current, measure voltage
- Measure voltage, no force

You can perform measurement operations at any time, even if you are not forcing current or voltage with the PPMU.

In the Digital Pattern Editor, navigate to the PPMU section of the Pin View pane to view voltage measurements. To programmatically make PPMU voltage measurements, use the following:

- niDigital PPMU Measure VI
- DigitalPpmu.Measure .NET method
- niDigital_PPMU_Measure C function

## Measure Current

You can perform the following current measurement operations with the PPMU:

- Force voltage, measure current
- Force current, measure current

You can only measure current while forcing voltage or current with the PPMU.

In the Digital Pattern Editor, navigate to the PPMU section of the Pin View pane to view current measurements. To programmatically make PPMU current measurements, use the following:

- niDigital PPMU Measure VI
- DigitalPpmu.Measure .NET method
- niDigital_PPMU_Measure C function

## Aperture Time

The PXIe-6571 supports discrete aperture times that are based on the sample rate of the ADC. You can configure aperture time on the PXIe-6571 to achieve a desired accuracy and/or speed.

When you use the NI-Digital Pattern Driver API or the Digital Pattern Editor Pin View pane to measure voltages and currents, the PXIe-6571 begins acquiring measurements immediately. The PXIe-6571 uses sampling ADCs to sample voltage and current. These ADCs also allow for the hardware averaging of acquired measurements using the aperture time. The number of hardware averages is represented by this equation:

***Number of Averages*** = (***Aperture Time***/4 µs)

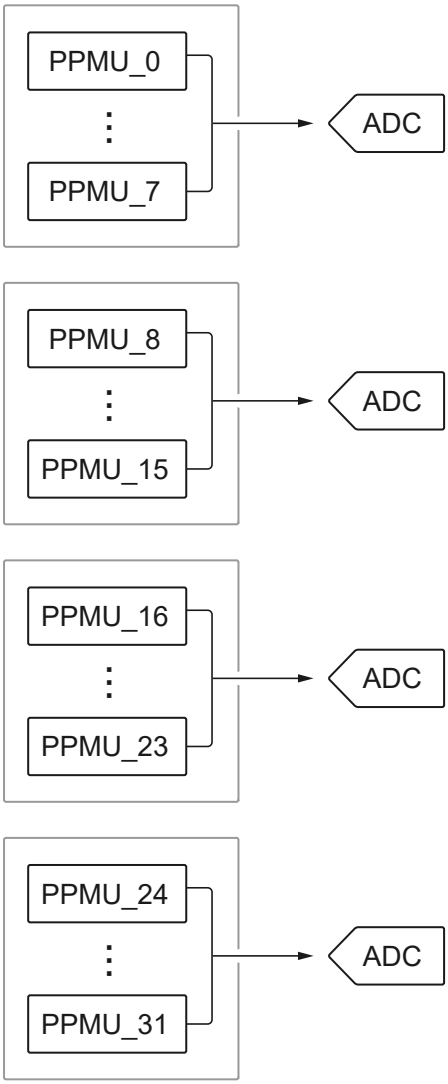
Refer to ***PXIe-6571 Specifications*** for more information about the relationship between measurement noise and selected aperture times.

To set the aperture time for your measurement, use the following:

- niDigital PPMU Configure Aperture Time VI
- DigitalPpmu.ConfigureApertureTime .NET method
- niDigital_PPMU_ConfigureApertureTime C function

## Optimizing PPMU Measurement Time

Each digital pattern instrument channel has dedicated per-pin PMU circuitry, but analog-to-digital conversion of the PPMU circuits is split into banks to increase operational efficiency: `DIO<0..7>`, `DIO<8..15>`, `DIO<16..23>`, and `DIO<24..31>`. An independent ADC is connected to each of these four banks of eight channels, meaning that PPMU current and voltage measurements can be run in parallel by taking measurements on channels in different banks.




When taking a measurement on two or more channels within a given bank, the connected ADC must be multiplexed between the selected channels. This results in an increase in measurement execution time that is dependent on the following characteristics:

- The number of PPMU measurement channels being multiplexed
- Whether the measurement channel is configured to make a voltage or current measurement
- Current measurement range

**Note** Measurement execution time is only dependent on the current measurement range if the channel is configured for current measurement.

The effect of measurement multiplexing on measurement execution time can be calculated using the following system of equations:

$$\text{Total Multiplexing Time Impact} = \left(\text{Number of Multiplexed Channels} * \text{Aperture Time}\right) + T_V + T_{HA} + T_{LA}$$

where

- $T_V = N_{VC} \times ST_V$

  where

  - **$T_V$**—Voltage Channel Settling Time Impact
  - **$N_{VC}$**—Number of Voltage Channels
  - **$ST_V$**—Voltage Channel Settling Time
- $T_{HA} = N_{HCC} \times ST_{HC}$

  where

  - **$T_{HA}$**—High Current Channel Settling Time Impact
  - **$N_{HCC}$**—Number of High Current Channels
  - **$ST_{HC}$**—High Current Channel Settling Time
- $T_{LA} = N_{LCC} \times ST_{LC}$

  where

  - **$T_{LA}$**—Low Current Channel Settling Time Impact
  - **$N_{LCC}$**—Number of Low Current Channels
  - **$ST_{LC}$**—Low Current Channel Settling Time

Refer to the following table for the settling time constants of different measurements.

**Table 10.** PPMU Measurement Settling Time Constants

| Measurement Settling Time | Constants | Note |
|---|---|---|
| Voltage Settling Time | 40 µs | — |
| High Current Settling Time | 130 µs | Applies to measurements in the 32 µA, 128 µA, and 2 mA ranges. |

| Measurement Settling Time | Constants | Note |
|---|---|---|
| Low Current Settling Time | 250 µs | Applies to measurements in the 2 µA range |

The settling time for multiplexing between two channels is determined by the channel being switched to, not the channel being switched from. Because of this, you do not have to factor in the channel you take your initial measurement on in a bank when calculating the impact of settling time on measurement execution time for that bank.

Measurement multiplexing has the highest impact on banks filled with low current measurement channels. For example, the highest possible multiplexing time impact for measurements taken with an aperture time of 1 µs is 1.76 ms:

(8 × 1 µs) + (7 × 250 µs) = 1.76 ms

The software overhead related to measurement multiplexing is the largest factor in measurement execution time for measurements with short aperture times. As aperture time increases, software overhead will have a progressively lower impact as measurement execution time comes to be dictated by the aperture time itself. Additionally, longer aperture times have a larger impact on measurement time if the measurements are taken on channels banked together and routed to the same ADC. Channels banked together share an ADC and need to be multiplexed, while channels routed to different banks are able to use unique ADCs and run in parallel. The following graphs demonstrate these relationships and show the improvement in execution time that running measurements in parallel on separate banks can provide.

**Figure 15.** PPMU Current Measurement (2 µA), Long Aperture Time

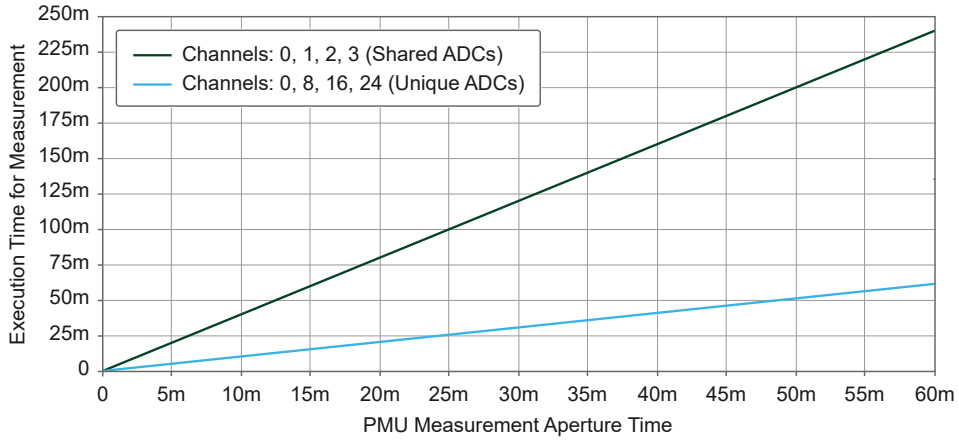


**Figure 16.** PPMU Current Measurement (2 µA), Short Aperture Time

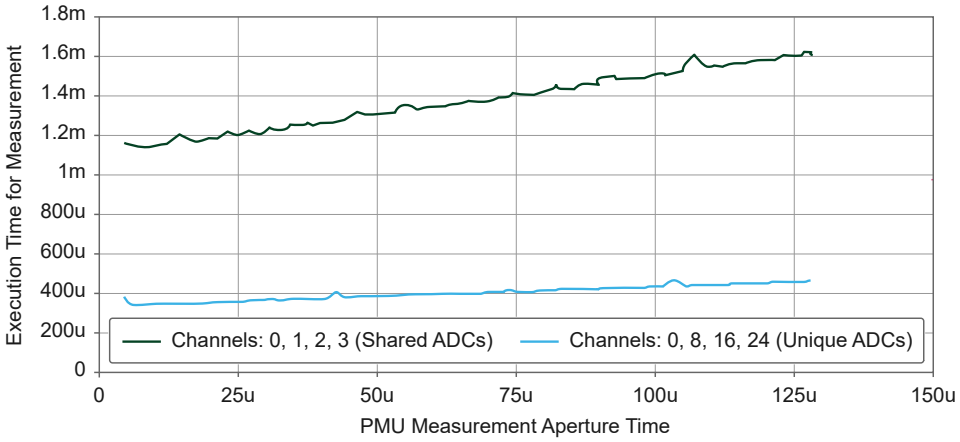



**Figure 17.** PPMU Voltage Measurement, Long Aperture Time

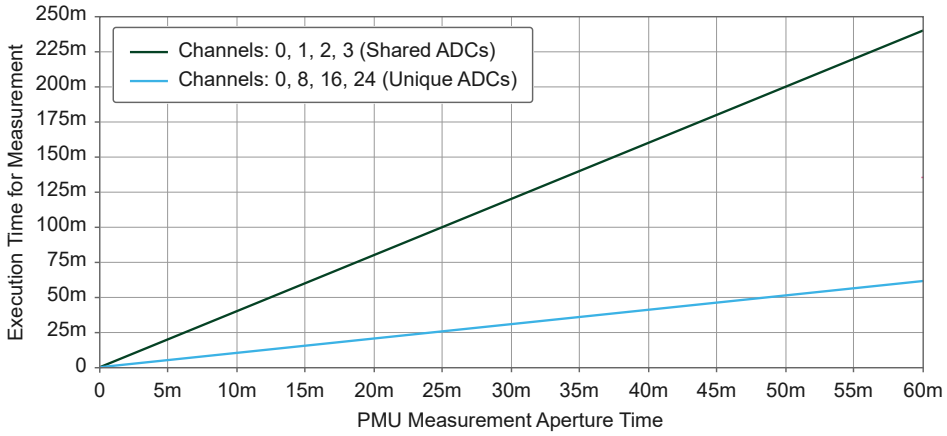
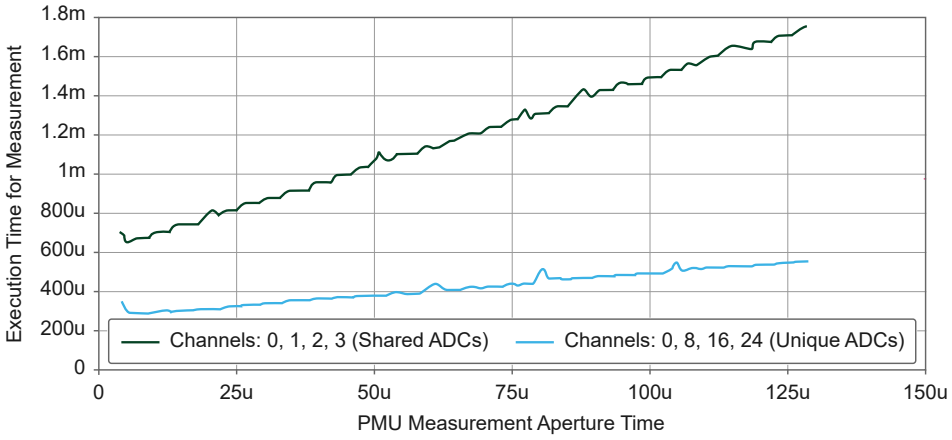



**Figure 18.** PPMU Voltage Measurement, Short Aperture Time




# Clocking

### Master Clock

The pattern execution engine, timing circuitry, and formatting circuitry on the digital pattern instrument run using a 100 MHz fixed master clock, MasterClk, which processes the data and places timing edges.

### Phase Lock Loop (PLL) Reference

The onboard frequency generator on the digital pattern instrument uses a phase-locked loop (PLL) circuit to lock the MasterClk of the instrument to the 100 MHz chassis backplane reference signal, PXIe_CLK100.

**Figure 19.** Digital Pattern Instrument Clocking Architecture




### Vector Periods

The time set specifies the vector period to use for pattern execution. You can modify vector periods on a vector-by-vector basis with the digital pattern instrument. Refer to the ***Digital Pattern User Manual*** for more information about specifying and loading time sets programmatically or through the Digital Pattern Editor.

**Related reference:**

- Synchronizing Multiple Instruments

**Related information:**

- NI-Digital Pattern User Manual

## Clock Generator

Use the Clock Generator API or the Clock feature on the Pin View pane of Digital Pattern Editor to generate a clock independent of the pattern. An active clock generator pin cannot be used in a pattern burst, but you may transition a pin from keep alive to clock generator.

# Sequencer Flags and Registers

## Sequencer Flags

Digital pattern instruments use sequencer flags for handshaking with the test program. You can set or clear any of the four sequencer flags in your pattern using the `set_seqflag` and `clear_seqflag` opcodes or programmatically in your test program. Use sequencer flags to control the execution flow of your patterns with the `jump_if` and `exit_loop_if` opcodes.

When you synchronize multiple digital pattern instruments, a backplane trigger line is automatically reserved for sequencer flag synchronization.

## Sequencer Registers

Pattern sequencer registers enable the pass of numeric values from the pattern sequencer to a test program using the `write_reg` opcode or control pattern repetition from the test program using the `repeat` and `set_loop` opcodes.

Refer to the ***Digital Pattern User Manual*** for more information about the NI-Digital Pattern Driver sequencer flags and registers API and opcodes.
**Related information:**

- NI-Digital Pattern User Manual

# Vector Memory

The NI-Digital Pattern Driver interprets patterns and automatically selects from the following types of memory to store vectors. A tiered memory structure enables low latency branching and large storage.

- **Fast Vector Memory (FVM)**—Stores the initial vectors after a sequencer branch for fastest access. The sequencer can branch to a non-sequential vector based on the evaluation of `jump`, `call`, or `looping` opcodes. The NI-Digital Pattern Driver loads all vectors with labels or vectors that use the `call` opcode in FVM.
- **Cache Vector Memory (CVM)**—Stores additional vectors shortly after a branch are

loaded in CVM.
- **Large Vector Memory (LVM)**—Stores vectors that do not require low latency branching.

If you run out of FVM and CVM, reduce the number of vector labels and `call` opcodes in the pattern. Alternatively, combine patterns, such as using source or capture functionality, to eliminate the need for unique patterns for each DUT register access.

If you run out of LVM, reduce the total number of vectors loaded in memory. This can often be accomplished through the use of `repeat`, `loop`, and `call` opcodes and source and capture functionality.



**Note** When you export a pattern name, that pattern starts to load vectors in FVM and CVM.

## Edge Multiplier

Edge multiplier is a time set and pattern file feature that enables up to two pin states per pin, per vector. Use the edge multiplier time set configuration to split the vector period into multiple DUT cycles for a pin.

Increasing the edge multiplier from 1 to 2 requires a single vector period to specify two pin states, one per DUT cycle. Setting the edge multiplier to 2 can achieve data rates greater than the base vector rate. Regardless of the edge multiplier of any pin, each vector only has only one time set and one opcode and consumes one vector's worth of memory.

An edge multiplier of 2 can be combined with non-return, return to low, and return to high drive formats to achieve higher bit rates. Surround by complement is not supported. A time set with an edge multiplier of 2 has three additional programmable edges that define the behavior of the second DUT cycle: data2, return2, and strobe2. Edge placement must not violate the minimum pulse width specification of the digital pattern instrument.

The edge multiplier of a pin may change on a per-vector basis. Use the .edge_multiplier statement to change the edge multiplier of each pin in a vector. The

time set and pattern vector must have the same edge multiplier per pin. Two pin states must be specified for any pin that has an edge multiplier of 2. The two pin states are limited to the following combinations: 00, 01, 10, 11, LL, LH, HL, HH, XX, MM, VV, DD, EE, and --.

## Shared Pins

System pins and DUT pins are two main types of pins.

- **System Pins**—Use for static control of system-level circuits. These pins have limited functionality and cannot be used during pattern bursts.
- **DUT Pins** —Use for one channel mapping (from the DUT pin to an instrument channel) per site. Each instrument channel can be associated with a single site or shared across multiple sites. Sharing across multiple sites could look like a shared circuit on a load board that interfaces to multiple sites. If multiple sites have the same channel mapping for the same pin, this is referred to as a shared pin.

In the following example, PinA is mapped to channel 0 for both sites 0 and 1, and mapped to channel 1 for sites 2 and 3.

```
<Connections>
<Connection pin="PinA" siteNumber="0,1" instrument="Dig" channel="0" />
<Connection pin="PinA" siteNumber="2,3" instrument="Dig" channel="1" />
 </Connections>
```

Any configuration of a shared pin is applied to all sites that are mapped to the same channel. In the previous example pin map, configuring $V_{OL}$ for site0/PinA applies the value to both site0/PinA and site1/PinA. When configuring a shared pin, you need to configure only one of the sites that is mapped to the shared channel. If you do not specify a site when configuring a shared pin, the value applies to all sites for the pin, regardless of the channel, just as it is with any other DUT pin.

Measurements taken on shared pins are returned per site, just as they are for any other DUT pin. In the previous example pin map, taking a PPMU measurement on PinA returns four results, one for each site. The results are identical for all the sites that are mapped to the same channel. For example, a single value is returned for measurements taken on site0/PinA or site1/PinA.

## Considerations When Using Shared Pins

When bursting a pattern that contains a shared pin, the physical channel that is mapped to the shared pin is active in the burst if at least one site of the shared pin is enabled and included in the burst. In the previous example pin map, bursting a pattern that contains PinA on either site0 or site1 causes channel 0 to be active in the burst.

The burst results for a shared pin are included in all sites that the shared pin is mapped to. Capture waveform data and History RAM results for shared pins are present in the samples for all sites the pin is shared on.

A failure on a shared pin is reported on all sites of the shared pin. In the previous example pin map, a failure on site0/PinA or site1/PinA causes both sites to report the failure.

When using shared pins in a source memory site unique waveform, the waveform must send the same data for all sites that have the same channel mapping.

When working with shared pins, call the Get Pin Results Pin Information API with pin lists instead of channel lists. NI-Digital Pattern Driver returns an error if the Get Pin Results Pin Information API is called with a channel that is shared on multiple sites.

# TDR

Use time domain reflectometry (TDR) to compensate for skew that load boards or cables introduce.

Use the NI-Digital Pattern TDR API or the TDR functionality of the Digital Pattern Editor to measure and deskew additional propagation delay of each channel that lies beyond the point at which the system timing is externally calibrated. Run the TDR measurement after changing cabling, instrument connections, or load boards.

**Note** For TDR to operate properly, ensure that the DUT sockets for the specified channels are empty when running the TDR measurement to open or shorted to ground when running the TDR measurement to short. Also ensure that the cables and load board connected to the digital pattern

> instrument make a 50 Ω controlled impedance system.

# History RAM

History RAM is onboard memory that stores cycle information during a pattern burst. Use the NI-Digital History RAM API or the History RAM feature of the Digital Pattern Editor to debug pattern execution.

History RAM cycle information includes:

- Pattern Name
- Time Set Name
- Vector Number
- Cycle Number
- Expected Pin Data
- Actual Pin Data
- Per-Pin Pass/Fail
- Scan Cycle Number

History RAM can be configured to acquire a finite number of samples or acquire samples continuously. When configured for finite acquisition, History RAM samples can only be fetched after the pattern burst has completed. When configured for continuous acquisition, History RAM samples can be fetched during a pattern burst.

History RAM information is only valid for the most recent pattern burst. Changing History RAM configuration, applying a timing sheet, unloading time sets, unloading patterns, or bursting a new pattern clears the previously acquired cycle information in History RAM.

## History RAM Configuration

Before bursting a pattern, you must configure the History RAM trigger and specify which cycles to acquire.

History RAM acquisition begins on one of the following triggers:

- First Failure
- Pattern Label
- Cycle Number

The cycles to acquire can be configured with the following niDigital properties:

- `History RAM:Cycles to Acquire Property`
- `History RAM:Pretrigger Samples Property`
- `History RAM:Max Samples To Acquire Per Site Property`
- `History RAM:Number of Samples is Finite Property`
- `History RAM:Buffer Size Per Site Property`

The onboard History RAM can acquire up to 8,192 samples of cycle information divided evenly across all configured sites.

### History RAM Streaming

When History RAM is configured for continuous acquisition, these samples will be streamed to the host buffer during pattern execution. This will free space in the onboard storage for additional samples.

If you do not acquire quickly enough, the host memory buffer will fill up and then any additional samples will cause an overflow. When streaming, you can always fetch the number of samples corresponding to the sample size specified in the History RAM Buffer Size Per Site property before an overflow error is possible. All samples fetched prior to an overflow are continuous and in order from the start of acquisition.

You can use the NI-Digital History RAM Streaming examples, located in the `<Public Documents>\National Instruments\NI-Digital\Examples` directory to learn how to use the History RAM streaming feature.

## Thermal Protection

The PXIe-6571 includes protections against excessive temperatures.

The digital pattern instrument powers down and disconnects the pin electronics in the presence of excessive heat. To prevent thermal protection from engaging, use the

digital pattern instrument within its specified ambient temperature range and ensure that the PXI chassis provides adequate airflow. Refer to ***PXIe-6571 Specifications*** for more information about the ambient temperature range.

**Note** The 32-channel PXIe-6571 requires an 82 W power and cooling–capable chassis; the 8-channel PXIe-6571 requires a ≥58 W power and cooling–capable chassis.

If the digital pattern instrument powers down, you will be notified with an error message in one of the following ways:

- NI-Digital Pattern Driver returns an error when you use any of the functions that program the hardware or check hardware status.
- NI hardware configuration software—Hardware Configuration Utility or MAX—returns an error if you run a self-test on your digital pattern instrument after it exceeds the thermal shutdown temperature. The thermal shutdown error continues to be reported until the instrument is successfully reset.
- The Active LED on the front panel of the digital pattern instrument glows red.

Before you reset the digital pattern instrument after thermal shutdown, cool the instrument to an acceptable range and resolve the environmental condition that caused the shutdown. Reset the PXIe-6571 by calling the niDigital Reset Device VI, the DigitalDriverUtility.Reset .NET method, or the niDigital_reset C function, performing an instrument reset in Hardware Configuration Utility or MAX, or power cycling the digital pattern instrument.

**Related information:**

- PXIe-6571 Specifications

## Calibration

Calibration operations compare the values a measuring instrument or measuring system returns to corresponding values from external standards. You can use calibration results to determine the measurement error and correct the error in the adjustment process. Digital pattern instruments support external calibration and self-

calibration.

Refer to ***PXIe-6571 Specifications*** for more information about accuracy and recommended calibration intervals.

**Related information:**

- PXIe-6571 Specifications

## External Calibration

Use NI Calibration Executive to automate the external calibration process. This is the only option to perform external calibration on PXIe-6571.

External calibration requires using high-precision calibrating instruments to verify and adjust the digital pattern instrument you are calibrating. Because the external calibration procedure changes calibration constants during adjustment, it invalidates any prior calibration certificate. If an external calibration is performed with traceable instrumentation, a new calibration certificate can be issued.

Visit ni.com/calibration for more information about NI calibration services and how to obtain the calibration certificate for your instrument.

### Calibration Interval

The calibration interval is the maximum recommended amount of time between external calibrations. Refer to ***PXIe-6571 Specifications*** for information about the calibration interval of your digital pattern instrument.

## Self-Calibration

You should self-calibrate the digital pattern instrument to compensate for temperature and environmental effects on the accuracy of the instrument since the last self-calibration. All self-calibration constants are calculated and stored in nonvolatile instrument memory.

The NI-Digital Pattern Driver provides a self-calibration API for analog calibration of the PXIe-6571. Use the niDigital Self Calibrate VI, the NIDigital.SelfCalibrate .NET method,

or the niDigital_SelfCalibrate C function when performing self-calibration. Refer to the API reference section in ***Digital Pattern User Manual*** for more information about self-calibration.

## Self-Calibration Considerations

- Self-calibrating the instrument resets the hardware and blocks access to the instrument for 3 to 5 minutes. Do not close the software executing the self-calibration and do not disable the instrument during this period.
- The digital pattern instrument opens switches on each channel during self-calibration to prevent damage to the DUT and to ensure correct calibration. You might encounter small voltage noise coupling from the digital pattern instrument through attached cabling during the calibration process.
- If cables remain connected during self-calibration, ensure that no fast transient signals, such as digital drivers, frequency generators, or noise sources, drive into the digital pattern instrument. Disconnect cables from the instrument during self-calibration if fast transient signals are otherwise unavoidable.

## Supported Self-Calibrated DC Features

All DC features of the digital pattern instrument are calibrated relative to an onboard precision reference that is valid only when the instrument is within its external calibration interval. Self-calibration compensates multiple voltage and current circuits for the pin driver, comparator, active load, and PPMU to correct for temperature and environmental effects.


© 2025 National Instruments Corporation.

**Artisan Technology Group** is an independent supplier of quality pre-owned equipment

## Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

## We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

## Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP