United Electronic Industries

The High-Performance Alternative

DNx-DIO-449

—

User Manual

High Voltage 48-Channel Digital Input Board
with Analog Readback and Guardian Functionality
for the PowerDNA Cube and RACK chassis

August 2018

PN Man-DNx-DIO-449

See the UEI website for complete terms and conditions of sale:
http://www.ueidaq.com/cms/terms-and-conditions/

CE

Contacting United Electronic Industries

Mailing Address:

27 Renmar Avenue
Walpole, MA 02081
U.S.A.

For a list of our distributors and partners in the US and around the world, please contact a member of our support team:

Support:

Telephone: (508) 921-4600
Fax: (508) 668-2350

Also see the FAQs and online “Live Help” feature on our web site.

Internet Support:

Support: support@ueidaq.com
Web-Site: www.ueidaq.com
FTP Site: ftp://ftp.ueidaq.com

Product Disclaimer:

WARNING!

DO NOT USE PRODUCTS SOLD BY UNITED ELECTRONIC INDUSTRIES, INC. AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS.

Products sold by United Electronic Industries, Inc. are not authorized for use as critical components in life support devices or systems. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Any attempt to purchase any United Electronic Industries, Inc. product for that purpose is null and void and United Electronic Industries Inc. accepts no liability whatsoever in contract, tort, or otherwise whether or not resulting from our or our employees' negligence or failure to detect an improper purchase.

Specifications in this document are subject to change without notice. Check with UEI for current status.

Table of Contents

List of Figures

Chapter 1 Introduction

This document outlines the feature set and use of the DNx-DIO-449 boards. DIO-449 boards are high-voltage digital input interface modules for the PowerDNA Cube, RACKtangle HalfRACK, and FLATRACK chassis.

This chapter includes the following sections:

- Organization of Manual (Section 1.1)
- Manual Conventions (Section 1.2)
- DIO-449 Board Overview (Section 1.3)
- Features (Section 1.4)
- Indicators (Section 1.5)
- Specification (Section 1.6)
- Device Architecture (Section 1.7)
- Gain Settings (Section 1.8)
- Connectors and Wiring (Pinout) (Section 1.9)
- PowerDNA Explorer for DIO-449 (Section 1.10)

1.1 Organization of Manual

This DIO-449 User Manual is organized as follows:

- **Introduction**
  This chapter provides an overview of DIO-449 high-voltage digital input board features, device architecture, connectivity, and logic.
- **Programming with the High-Level API**
  This chapter provides an overview of how to create a session, configure the session, and interpret results on the DIO-449 series boards for programming with the high-level Framework.
- **Programming with the Low-Level API**
  This chapter provides an overview of low-level API commands for configuring and using the DIO-449.
- **Appendix A - Accessories**
  This appendix provides a list of accessories available for use with the DIO-449 boards.
- **Appendix B - Programming Guide for GND/VDD Open Contact Monitoring**
  This appendix provides a programming guide for monitoring GND/VDD open contact switched inputs.
- **Index**
  This is an alphabetical listing of the topics covered in this manual.

1.2 Manual Conventions

To help you get the most out of this manual and our products, please note that we use the following conventions:

Tips are designed to highlight quick ways to get the job done or to reveal good ideas you might not discover on your own.

NOTE: Notes alert you to important information.

CAUTION! *Caution advises you of precautions to take to avoid injury, data loss, and damage to your boards or a system crash.*

Text formatted in **bold** typeface generally represents text that should be entered verbatim. For instance, it can represent a command, as in the following example: "You can instruct users how to run setup using a command such as **setup.exe**."

Bold typeface will also represent field or button names, as in "Click **Scan Network**."

Text formatted in `fixed` typeface generally represents source code or other text that should be entered verbatim into the source code, initialization, or other file.

Before you begin:

No HOT SWAP

Always turn POWER OFF before performing maintenance on a UEI system. Failure to observe this warning may result in damage to the equipment and possible injury to personnel.

Usage of Terms

Throughout this manual, the term "Cube" refers to either a PowerDNA Cube product or to a PowerDNR RACKtangle™ rack mounted system, whichever is applicable. The term DNR is a specific reference to the RACKtangle, DNA to the PowerDNA I/O Cube, and DNx to refer to both.

1.3 DIO-449 Board Overview

The DIO-449 boards are 48 channel, high performance AC and DC digital input boards designed for use in a wide variety of digital monitoring applications.

The DNA-DIO-449, DNR-DIO-449, and DNF-DIO-449 boards are compatible with the UEI Cube, RACKtangle, and FLATRACK chassis respectively. All board versions are functionally identical except for the mounting hardware. The DNA version is designed to stack in a Cube chassis. The DNR/F versions are designed to plug into the backplane of a RACK chassis.

1.3.1 Reading Digital Output Bits

The digitized output bits for each of the 48 channels are available in two 24-bit ports (channels 23 to 0 can be read from port 0, and channels 47 to 24 can be read from port 1). Each port provides the data in a 24-bit read, which simplifies programming and maximizes throughput.

The DIO-449 supports reading all 48 channel bits at rates up to 5120 samples per second. Automatic Change Of State (COS) detection is available with 200µs accuracy.

1.3.2 Guardian Diagnostics

The DIO-449 is equipped the “Guardian advantage,” which provides powerful diagnostic capabilities. When enabled, Guardian diagnostic circuitry can inject a test signal at the channel input, allowing the full input hardware chain to be tested for proper functionality. The DIO-449 can monitor switches or contacts without external circuitry using an on-board voltage supply. Since the input hardware for each channel includes an A/D converter in the chain, the actual raw analog voltage at each input can be monitored. This capability, combined with the board’s ability to switch a fixed reference voltage into each channel, allows a complete and reliable self-test of each channel. The capability of measuring raw analog voltages also provides quick and accurate detection of short and open circuits, as well as detection of marginal or failing drive circuitry. This analog input capability can be used as a powerful installation and diagnostic tool.

1.3.3 Hysteresis & Debouncing

The DIO-449 board also offers programmable logic thresholds and hysteresis over the full input range. Thresholds and hysteresis are independently programmable for each channel. The board additionally supports user programmable debouncing intervals which may be set for each channel independently with durations up to 409.5 ms.

1.3.4 Isolation & Over-voltage Protection

Each board provides 350 V_{rms} isolation between the I/O and the chassis and other installed I/O boards. All inputs are over-voltage protected from -350 to +350 VDC and against ESD.

1.3.5 Software Support

Software included with the DNx-DIO-449 provides a comprehensive yet easy to use API that supports all popular operating systems including Windows, Linux, real-time operating systems such as QNX, RTX, VXworks and more. The UEIDAQ Framework supplies complete support for developers creating applications in Windows based data acquisition software languages as well as Windows application packages such as LabVIEW, MATLAB/Simulink, or any application which supports ActiveX or OPC servers.

1.4 Features

The DIO-449 board provides the following features:

- ±150 VDC / 0-150 VAC input range
- Sample rate of 5120 samples per second
- Per-channel programmable
  - Input transition levels (hysteresis)
  - Input voltage level range (gain)
  - AC or DC mode
  - Debounce intervals
- Change of state detection with 200 μs timestamp resolution
- 350 V_{AC} isolation
- Contact monitoring without external components (open-circuit detection)
- Guardian-series diagnostics
  - Analog voltage measurement on each channel
  - Internal test signal injection for self-test

1.5 Indicators

The DIO-449 indicators are described in **Table 1-1** and illustrated in **Figure 1-1**.

Table 1-1 DIO-449 Indicators

LED Name	Description
RDY	Indicates board is powered up and operational
STS	Indicates which mode the board is running in: • **OFF**: Configuration mode (e.g., configuring channels, running in point-by-point mode) • **ON**: Operation mode (e.g., ACB mode)

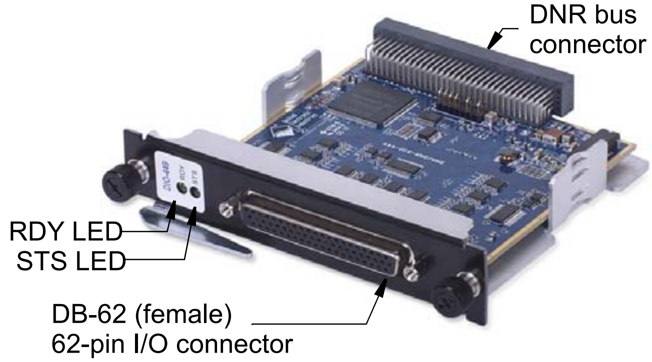


Figure 1-1. Photo of DIO-449 Digital Input Board

1.6 Specification

The technical specifications for the DIO-449 are listed in **Table 1-2**.

Table 1-2. DNx-DIO-449 Technical Specifications

Number of channels	48 digital inputs
Port configuration	Two 24-bit ports
Input range	-150 to +150VDC, 0 to 150 VAC (AC freq: 42 - 2500 Hz)
Input gains	x1 default. Gains of x2, x5 and x10 are provided if higher resolution is required for lower voltage input ranges
Input high voltage	Programmable from 0 to 150 VDC/VAC (default: 12 VDC, 60 VAC)
Input OFF voltage	Programmable from 0 to 150 VDC/VAC (default: 1.25 VDC, 15 VAC)
Hysteresis (voltage input)	Programmable, 0 to 150 VDC/VAC (default 10.25 VDC/45 VAC)
Input impedance	> 2.2 MegOhm
Input open circuit state	Programmable high or low via signal injection diagnostic stage. Each channel is independently programmable.
Input FIFO	1024 samples
Input Throughput Rate	5120 Hz
Change of state detection	Based on the change of one or more inputs.
COS timestamp accuracy	200 µS
Voltage measurement and threshold voltage accuracy	DC: ± 50 mV (-150 VDC to 150 VDC), AC: ± 150 mVAC (0 VAC to 150 VAC)
Input protection	±350 VDC and ESD
Input Isolation	350 Vrms
Power dissipation	2 W, max.
Operating Temp. Range	Tested -40 to +85 °C
Operating Humidity	95%, non-condensing
Vibration *IEC 60068-2-6* *IEC 60068-2-64*	5 g, 10-500 Hz, sinusoidal 5 g (rms), 10-500 Hz, broad-band random
Shock *IEC 60068-2-27*	50 g, 3 ms half sine, 18 shocks @ 6 orientations 30 g, 11 ms half sine, 18 shocks @ 6 orientations
MTBF	500,000 hours

1.7 Device Architecture

The DIO-449 supports 48 digital inputs with per-channel programmable logical "high" and "low" input voltage (hysteresis) within ±125 VDC and 0-150 VAC with 10-12 bit accuracy. The DIO-449 is similar to the DIO-448 but supports a wider range of digital/DC and analog/AC voltage inputs on a single board with more protection. The DIO-448 and DIO-449 are pin-compatible.

A diagram of the DIO-449 architecture is shown in **Figure 1-2**:

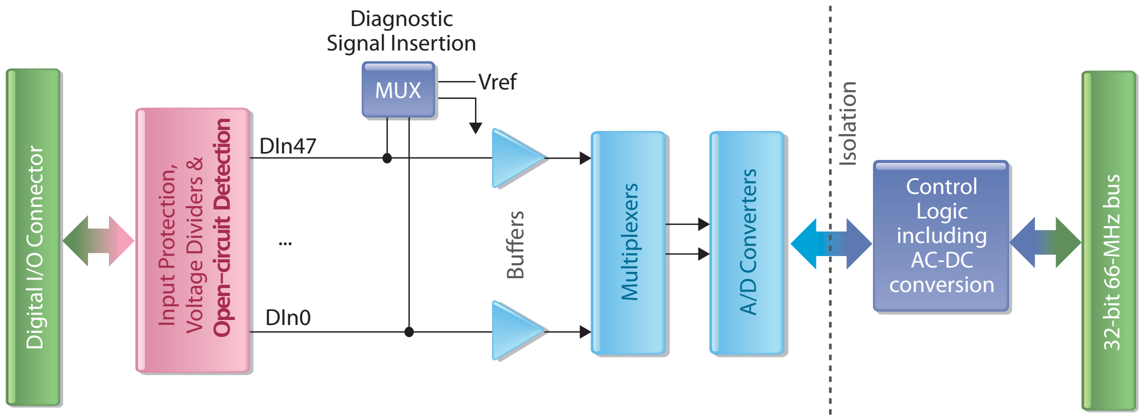


Figure 1-2. Architecture Block Diagram of DNA-DIO-449

1.7.1 Input Circuitry

The female DB-62 connector at the front of the DIO-449 provides access to 48 digital/analog inputs, which share the same isolated ground. Each input passes through the circuit depicted in **Figure 1-3**.

The first stage of circuitry for each of the input channels includes an optional open-circuit detection circuit, a voltage divider, and optional diagnostic signal insertion. The 10MΩ resistor and -15V source circuit are used for optional/ programmable open-circuit detection. The input is divided down to ADC levels by voltage division resistors, and the Guardian self-test circuit block containing a DAC and resistors can optionally be enabled for diagnostic signal insertion, which affects 4-channel groupings.

The input signal is buffered by an amplifier and passed through a multiplexer to reach the 1/2/5/10x gain buffer (per-channel selectable, Section 1.8) and 16-bit successive approximation (SAR) ADC.

Each input channel circuit is configured as shown in **Figure 1-3**:

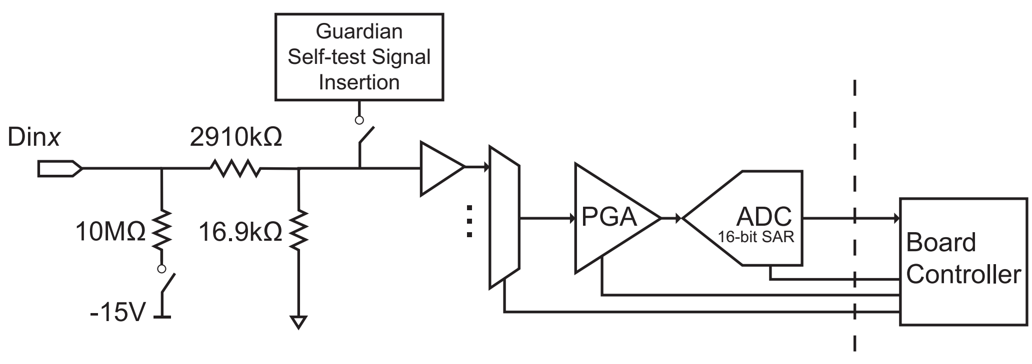


Figure 1-3. Abstracted Single Channel Diagram for DIO-449

The DIO-449 multiplexer is composed of three identical multiplexer blocks that support 16-channels each. Each multiplexer acquires one sample and transmits the sample across isolation circuitry to the board's logic controller. This means that every 16 channels are captured sequentially at a conversion clock of 81920Hz, and all 3 groups of 16 channels are complete at a rate of 5120Hz. The controller samples and processes three channels in parallel at all times.

1.7.2 Sample Averaging

The digitized analog output from the A/D converter is stored in a 256-sample snapshot buffer. There is one 256-sample snapshot buffer for each channel. The RMS (AC) or average (DC) voltage is calculated from the samples stored in the snapshot buffer, and each incoming sample triggers a re-calculation.

For AC inputs, the frequency of the input waveform must be under 2500Hz to avoid aliasing and over 42Hz to fill the buffer at least once every second. True RMS can be calculated for all data in the 256-sample buffer or for the last complete period only. Averaging all 256 samples or only the last complete period mode is a user-programmable mode. Using all periods in the buffer provides a more accurate reading for steady state signals but will delay detection of changes of state on a channel due to averaging.

1.7.3 Hysteresis & Debouncing

The averaged voltage (AC RMS or DC average) is then evaluated in the digital comparator circuitry, which compares the averaged voltage to the low and high threshold limits programmed for the channel hysteresis. Debouncing is optionally performed (up to 12-bit counter running from 10KHz clock). Debouncing delays can be programmed from 0.0 to 409.5 ms.

1.7.4 Reading Bit Data & Timestamps

After debouncing, the analog input signal for each of the 48 channels has been interpreted as a digital 0 or 1. The bit values for all 48 channels are stored as two 24-bit words, along with a timestamp: the two 24-bit words represent bit values for channels 23 to 0 (port 0) and bit values for channels 47 to 24 (port 1).

In continuous mode, port data and timestamp values are read upon request by the Cube or RACK CPU in response to a function call from the user application.

If change-of-state mode is enabled, change-of-state detection circuitry will issue an interrupt to the Cube or RACK CPU if the binary data for any channel transitions from a 0 to 1 or 1 to 0. The chassis CPU reads the data and timestamp from the DIO-449 and sends a packet containing port data, the change of state condition, and a timestamp to the host.

The user can request data from either of the two independent hardware buffers: the 1024-sample Input FIFO or the 256-sample snapshot buffer. Requesting data from the snapshot buffer will pause acquisition while the request is being completed.

1.8 Gain Settings

The DIO-449 is capable of operating within the following voltage ranges:

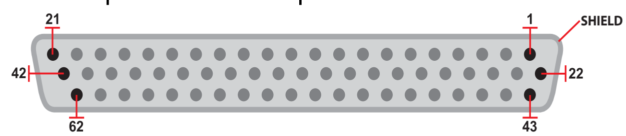
Gain Code	Gain	DC operating range	AC operating range
0	x1	$\pm 150V_{DC}$	$\pm 215V_{PP}$ (0-152V_{RMS})
1	x2	$\pm 107V_{DC}$	$\pm 107V_{PP}$ (0-76V_{RMS})
2	x5	$\pm 42V_{DC}$	$\pm 42V_{PP}$ (0-30V_{RMS})
3	x10	$\pm 21V_{DC}$	$\pm 21V_{PP}$ (0-15V_{RMS})

The signal must be kept within the operating range, otherwise the measured values returned by the DIO-449 will be incorrect:

- **DC**: applying signals larger than the voltage range will appear as the maximum value of the +/- voltage range to be read, (i.e., saturation or clipping returns the value of the positive or negative DC/DC power supply rail). Depending on the characteristics of your signal, it is possible to detect clipping in software by checking for the maximum value of the rail. For DC voltages, it is possible to apply more than 150VDC (up to 215V) without clipping; however, this will stress the components beyond the normal operating range and is not recommended.
- **AC**: applying signals outside of the voltage range will cause erroneously low RMS values to be calculated and will stress the components.

1.9 Connectors and Wiring (Pinout)

Figure 1-4 shows the pinout of the 62-pin female connector of the DIO-449:



Pin	*Signal*	*Pin*	*Signal*	*Pin*	*Signal*
1	Rsvd	22	Rsvd	43	Gnd
2	Rsvd	23	Gnd	44	Rsvd
3	Rsvd	24	Gnd	45	Gnd
4	Rsvd	25	Rsvd	46	DIn 47
5	DIn 45	26	DIn 46	47	DIn 44
6	DIn 42	27	DIn 43	48	DIn 41
7	DIn 39	28	DIn 40	49	DIn 38
8	DIn 36	29	DIn 37	50	DIn 35
9	DIn 33	30	DIn 34	51	DIn 32
10	DIn 30	31	DIn 31	52	DIn 29
11	DIn 27	32	DIn 28	53	DIn 26
12	DIn 24	33	DIn 25	54	DIn 23
13	DIn 21	34	DIn 22	55	DIn 20
14	DIn 18	35	DIn 19	56	DIn 17
15	DIn 15	36	DIn 16	57	DIn 14
16	DIn 12	37	DIn 13	58	DIn 11
17	DIn 9	38	DIn 10	59	DIn 8
18	DIn 6	39	DIn 7	60	DIn 5
19	DIn 3	40	DIn 4	61	DIn 2
20	DIn 0	41	DIn 1	62	Rsvd
21	Rsvd	42	Gnd		

Rsvd - Reserved, do not connect

Figure 1-4. DB-62 I/O Connector Pinout for DIO-449

NOTE: The DIO-449 inputs are numbered from DIn0 through DIn47.

1.10 PowerDNA Explorer for DIO-449

The PowerDNA Explorer right-hand pane of the DIO-449 display contains two tabs:

- Immediate: the runtime values currently in use
- Initialization: the values that the DIO-449 is set to at startup

1.10.1 Initialization Values

The Initialization tab, shown in **Figure 1-5** below, contains the following columns:

- **DInx**: read-only display of the channel number.
- **Name**: read-only display of the channel name.
- **DC**: when checked the voltage measurement algorithm will use direct current (DC). When unchecked it uses alternating current (AC_{RMS}).
- **Gain**: read-only display of the gain from “Gain Settings” on page 7.
- **Low Threshold**: threshold voltage at which analog voltages less than this value are interpreted as a binary 0. Units are VDC or VAC_{RMS}.
- **High Threshold**: threshold voltage at which analog voltages greater than this value are interpreted as binary 1. Units are VDC or VAC_{RMS}.

NOTE: Input values between the low and high threshold do not change the output binary state until a threshold is crossed.

- **Debounce**: user-programmable debounce period from 0 to 409.5 ms

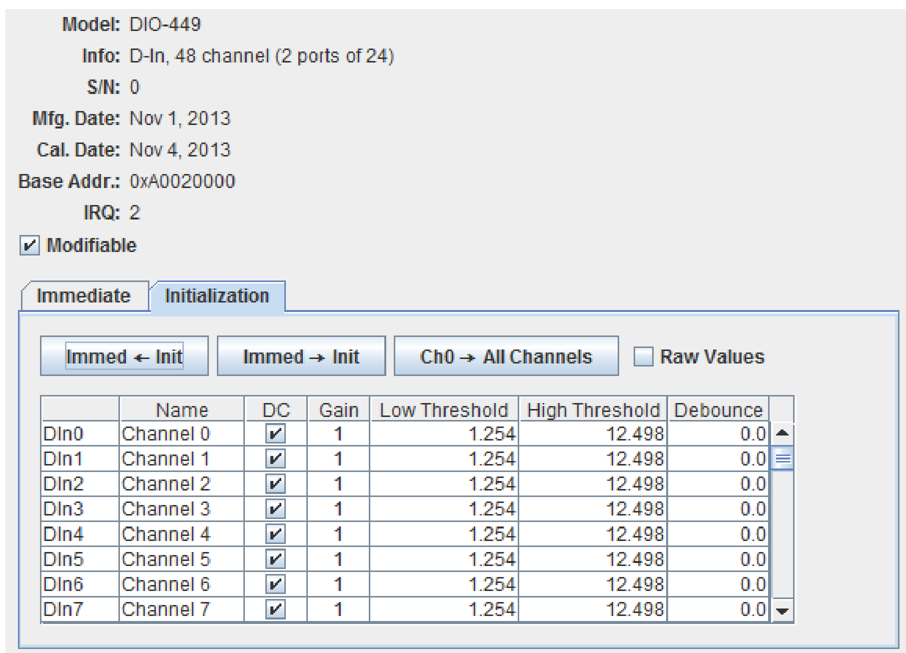


Figure 1-5. PowerDNA Explorer DIO-449 Immediate Pane

For convenience, the following additional buttons are available:

- **Immed** → **Init** or **Immed** ← **Init**: copies values between panes.
- **Ch0** → **All Channels**: copies channel 0 to all other channels.
- **Raw Values**: displays raw values as retrieved from non-volatile memory.

Initialization values are stored in non-volatile memory on the DIO-449 board and are loaded at startup. You can change any of the factory default values and save them into the non-volatile memory on that specific DIO-449 by clicking the **Store Config** button. Stored values will be retained if the DIO-449 board is removed and installed into different IOM (Cube or RACKtangle).

1.10.2 Immediate Values

The Immediate tab, shown in **Figure 1-6** below, contains the following columns:

- **DIn*x***: read-only display of the channel number.
- **Name**: channel name that can be updated and stored.
- **DIn**: read-only display of the digital/binary interpretation of the voltage.
- **AIn**: read-only display of the analog voltage. Units are VDC or VAC_{RMS}.
- **DC**: the voltage measurement algorithm as DC or AC.
- **Gain**: read-only display of the gain from “Gain Settings” on page 7.
- **Low Threshold**: threshold voltage in which analog voltages less than this value are interpreted as a binary 0. Units are VDC or VAC_{RMS}.
- **High Threshold**: threshold voltage in which analog voltages greater than this value are interpreted as a binary 1. Units are VDC or VAC_{RMS}.

NOTE: Debounce values are not displayed in the Immediate pane.

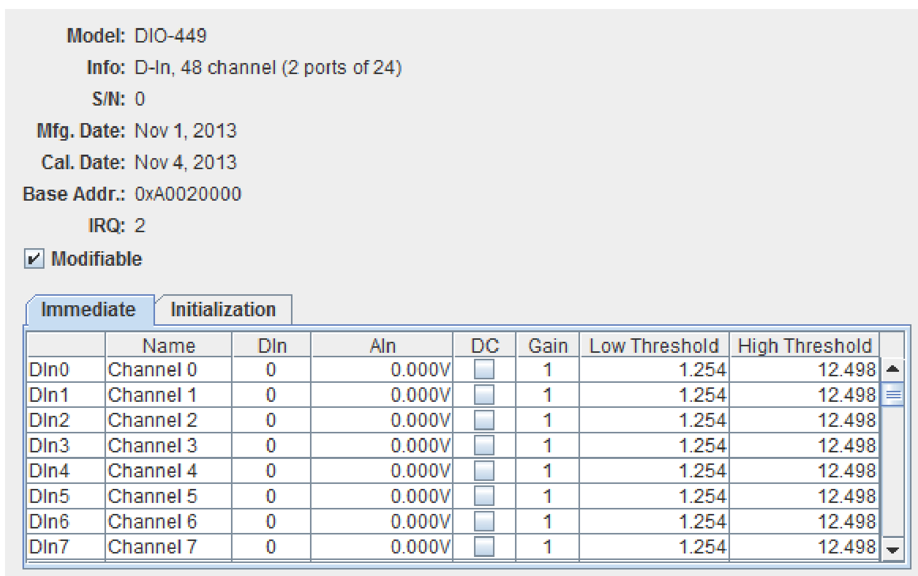


Figure 1-6. PowerDNA Explorer DIO-449 Immediate Pane

The Immediate display shows the current runtime values used to interpret the analog input voltage (shown in the AIn column) as a digital 1 or 0 (shown in the DIn column).

At system start up or after a reset, the DIO-449 initial values shown in the Immediate pane are the same as the values shown in the Initialization pane. You can update the low threshold or high threshold values in the Immediate pane, which will store them in volatile memory until the system is powered down.

To show the present values for AIn and DIn, click the **Start Reading Input Data** button.

Chapter 2 Programming with the High-Level API

This chapter provides the following information about using the UeiDaq Framework High-Level API to control the DIO-449:

- Creating a Session (Section 2.1)
- Configuring the Resource String (Section 2.2)
- Configuring Digital Inputs (Section 2.3)
- Configuring the Timing (Section 2.4)
- Reading Data (Section 2.5)
- Cleaning-up the Session (Section 2.6)

UeiDaq Framework is object oriented and its objects can be manipulated in the same manner from different development environments, such as Visual C++, Visual Basic, or LabVIEW.

The following section focuses on the C++ API, but the concept is the same no matter what programming language you use.

Please refer to the “UeiDaq Framework User Manual” for more information on use of other programming languages.

2.1 Creating a Session

The Session object controls all operations on your PowerDNx device. The first task when programming using the high-level Framework is to create a session object:

```
// create a session object

CUeiSession session;
```

2.2 Configuring the Resource String

UeiDaq Framework uses resource strings to select which device, subsystem and channels to use within a session. The resource string syntax is similar to a web URL:

```
<device class>://<IP address>/<Device Id>/<Subsystem><Channel list>
```

For PowerDNA and RACKtangle, the device class is **pdna**.

For example, the following resource string selects digital input lines 0,1 on device 1 at IP address 192.168.100.2: “pdna://192.168.100.2/Dev1/Di0,1”

2.3 Configuring Digital Inputs

The DIO-449 is an AC and DC digital input device that supports Industrial input voltage ranges of ±150 VDC and 0-150 VAC.

When programming the DIO-449, note that digital channels correspond to a physical port on the device. Port 0 corresponds to channels 23 through 0, and port 1 corresponds to channels 47 through 24. You cannot configure a session that only accesses a subset of channels within a digital port.

In general, Framework sessions are unidirectional. Since the DIO-449 is also unidirectional, you only need to configure one session.

Use the method **CreateDIIndustrialChannel()** to configure digital input parameters, such as setting threshold levels and enabling digital filters to eliminate glitches and spikes.

The following example code shows the configuration of the digital input ports (port 0 and port 1) of the DIO-449 set as device 1:

```
// Configure session to write to ports 0 to 1 on device 1

session.CreateDIIndustrialChannel("pdna://192.168.100.2/Dev1/Di0:1",
                                  1.5,
                                  5.0,
                                  1.0);
```

The example code shows the configuration of the following parameters:

- **Low Threshold**: the low hysteresis threshold.
- **High Threshold**: the high hysteresis threshold.
- **Digital input filter (debouncer)**: the minimum pulse width in ms. Use 0.0 to disable debouncing.

The following DIO-449 advanced features are accessible through channel object methods (or property nodes under LabVIEW):

- **AC/DC mode**: in AC mode, the RMS voltage measured at each input is compared with low and high thresholds to determine the state of the input. In DC mode, an average (DC) voltage is calculated, which is compared with low and high thresholds to determine the state of the input. AC mode can be enabled using the **EnableACMode()** method.
- **Signal injection**: state of Guardian multiplexer that injects a diagnostic test voltage at the channel input. The injection mode can be set using the **SetMux()** method.
  - **Tristate**: disconnects the internal test voltage source from the input channel, (i.e., set `mux` parameter to `UeiDigitalInputMuxTriState`).
  - **Diagnostic**: connects the internal voltage source directly to the input line, resulting in a known voltage level at the channel input for test purposes, (i.e., set `mux` parameter to `UeiDigitalInputMuxDiag`).
- **Voltage Supply**: programmable voltage level applied by the internal test source. The voltage supply level can be set using the **SetVoltageSupply()** method.
- **Input Gain**: the input gain used by the ADC when measuring the input voltage. The input gain can be set using the **SetInputGain()** method.

The following code provides an example of setting DIO-449 input line parameters:

```
// The DIO-449 input data is organized in two ports of 24 lines

for(int p=0; p<diSs.GetNumberOfChannels(); p++)
{
   CUeiDIIndustrialChannel* pDIIndusChan =
        dynamic_cast<CUeiDIIndustrialChannel*>(diSs.GetChannel(p));
   for(int l=0; l < diSs.GetDevice()->GetDIResolution(); l++)
   {
      pDIIndusChan->SetLowThreshold(l, 4.0);
      pDIIndusChan->SetHighThreshold(l, 5.0);
      pDIIndusChan->SetMinimumPulseWidth(l, 10);
      pDIIndusChan->SetVoltageSupply(l, 1.0);
      pDIIndusChan->SetMux(l, UeiDigitalInputMuxDiag);
   }
}
```

To read the voltage and current flow through each of the digital input lines, you must program the DIO-449 as you would an analog input device and create a new analog input session (different from the one you are using to read the digital states of the input lines).

The following code provides an example of configuring the input lines to be read from:

```
// Create an AI session

CUeiSession aiSession

// Read voltage from all 48 lines. Only the first parameter “resource”
// is used with the DIO-449, the other parameters are ignored.

aiSession.CreateAIChannel(“pdna://192.168.100.2/Dev1/Ai0:47”,
                  -10, 10,UeiAIChannelInputModeDifferential);
```

2.4 Configuring the Timing

You can configure the DIO-449 to run in simple mode (point by point), buffered mode (ACB mode), or DMAP mode.

- In simple mode, the delay between samples is determined by software on the host computer.
- In DMAP mode, the delay between samples is determined by the DIO-449 on-board clock. Data is transferred one scan at a time between PowerDNA and the host PC.
- In buffered ACB mode, the delay between samples is determined by the DIO-449 on-board clock. Data is transferred in blocks between PowerDNA and the host PC.

The following code shows how to configure the simple mode. Please refer to the “UeiDaq Framework User’s Manual” to learn how to use other timing modes.

```
// configure timing for point-by-point (simple mode)

session.ConfigureTimingForSimpleIO();
aiSession.ConfigureTimingForSimpleIO();
```

2.5 Reading Data

Reading analog and digital data from the DIO-449 is done using reader objects.

The following sample code shows how to create a scaled reader object and read samples:

```
// create a reader and link it to the digital session's stream
CUeiDigitalReader diReader(session.GetDataStream());
// create a reader and link it to the analog session's stream
CUeiAnalogScaledReader aiReader(aiSession.GetDataStream());
// read one digital scan, the buffer must be
// big enough to contain one value per port
uInt32 data[2];
reader.ReadSingleScan(data);
// read voltages from all input lines
double volts[48];
aiReader.ReadSingleScan(volts);
```

2.6 Cleaning-up the Session

The session object will clean itself up when it goes out of scope or when it is destroyed. To reuse the object with a different set of channels or parameters, you can manually clean up the session as follows:

```
// clean up the session
session.CleanUp();
aiSession.CleanUp();
```

Chapter 3 Programming with the Low-Level API

This chapter provides the following information about using the low-level API to program the DIO-449:

- Low-Level API Functions (Section 3.1)
- Low-Level Code Examples (Section 3.2)

The low-level API offers direct access to PowerDNA DaqBIOS protocol and also allows direct access to device registers.

However, when programming in Windows OS, we recommend that you use the UeiDaq Framework High-Level API (see **Chapter 2**) because it is easier to use. You should only need to use the low-level API if you are using an operating system other than Windows.

For additional information regarding low-level programming, refer to the "PowerDNA API Reference Manual" located in either directory:

- For Linux systems:
  <extraction root>/<PowerDNA-x.y.z>/Documentation
- For Windows systems:
  Start » All Programs » UEI » PowerDNA » Documentation

3.1 Low-Level API Functions

Table 3-1 lists low-level functions specific to the DIO-449. Refer to the "PowerDNA API Reference Manual" for detailed information regarding these functions, as well as other low-level functions.

Table 3-1DIO-449 Low-level API Functions

Function	Description
`DqAdv449ConfigEvents`	Configures events in which program firmware sends asynchronous event notification packets when user-specified conditions are met
`DqAdv449GetModeGainLevels`	Retrieves state of AC/DC configuration, gain setting, and logic levels for all 48 input lines
`DqAdv449Read`	Reads the current state of digital inputs, which are mapped to two digital ports, port 0 and port 1. The 24 LSBs of port 0 map to the bit states for channels 23 thru 0, and the 24 LSBs of port 1 map to the bit states for channels 47 thru 24
`DqAdv449ReadAdc`	Reads raw conversion results for each channel specified in the channel list and stores results alongside converted data
`DqAdv449ReadBlock`	Reads a block of the 256 most recent ADC raw conversion results for the specified channel. The 257th value is the gain setting, which determines the scaling factors to be applied to the data

Table 3-1DIO-449 Low-level API Functions (Cont.)

Function	Description
`DqAdv449SetAveragingMode`	Sets the mode of averaging digital inputs. Averages can be calculated by averaging all periods in a 256-sample buffer or by averaging only the samples from the last complete period
`DqAdv449SetDebouncer`	Sets the debouncing delay time for the digital comparator circuitry for channels specified in the channel list
`DqAdv449SetGDacs`	Sets Guardian DAC input voltages to be injected on any input lines that are enabled by the `DqAdv449SetGMux` function
`DqAdv449SetGMux`	Sets mux enables for connecting Guardian test circuitry to the 48 input lines. Mux enables that connect Guardian DACs to the input lines affect groupings of 4 channels. Mux enable that disconnects -15V power source affects all 48 channels
`DqAdv449SetLevels`	Sets gain, mode, and high and low threshold levels

3.2 Low-Level Code Examples

Application developers are encouraged to explore the existing source code examples when first programming the DIO-449. Sample code provided with the installation is self-documented and serves as a good starting point.

Code examples are located in the following directories:

- On Linux systems: <PowerDNA-x.y.z>/src/DAQLib_Samples
- On Windows: *Start » All Programs » UEI » PowerDNA » Examples*

Appendix A

A. Accessories

The following cables and STP boards are available for the DIO-449 board.

DNA-CBL-62

This is a 62-conductor round shielded cable with 62-pin male D-sub connectors on both ends. It is made with round, heavy-shielded cable; 2.5 ft (75 cm) long, weight of 9.49 ounces or 269 grams; up to 10ft (305cm) and 20ft (610cm).

DNA-STP-62

The STP-62 is a Screw Terminal Panel with three 20-position terminal blocks (JT1, JT2, and JT3) plus one 3-position terminal block (J2). The dimensions of the STP-62 board are 4w x 3.8d x1.2h inch or 10.2 x 9.7 x 3 cm (with standoffs). The weight of the STP-62 board is 3.89 ounces or 110 grams.

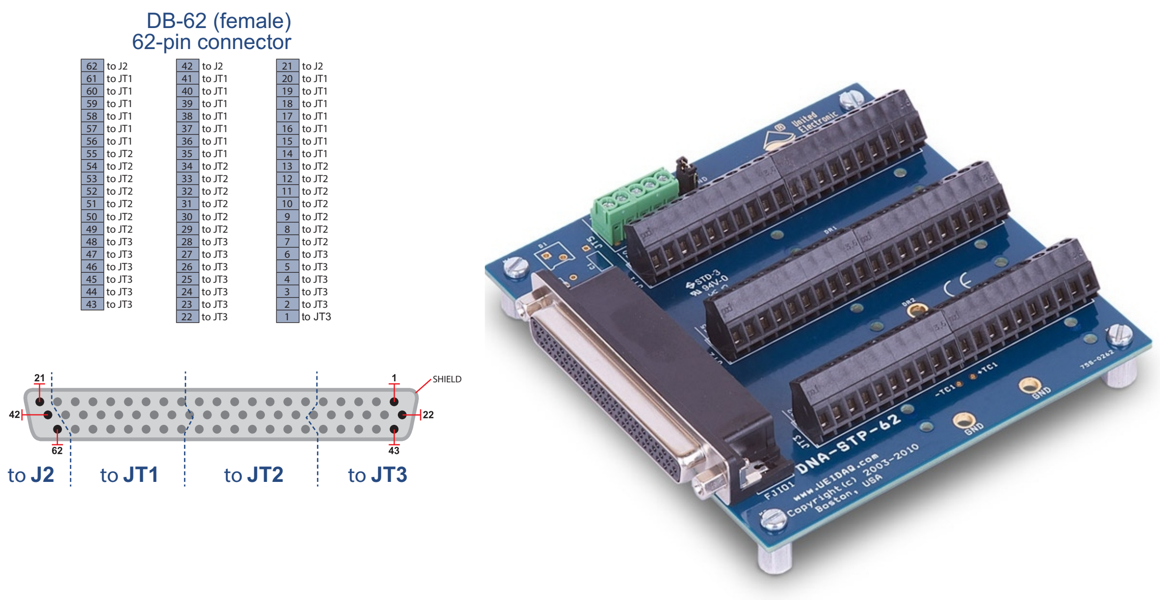


Figure A-1. Pinout and Photo of DNA-STP-62 Screw Terminal Panel

Appendix B

The following sections provide a programming guide for GND and VDD open contact monitoring.

B.1 GND Open Contact Monitoring

Figure B-1 shows an example of sensor/switch output circuitry and DIO-449 input circuitry for monitoring GND open contact switches.

In this configuration, the -15V DIO-449 on-board voltage source is connected to the Din_x input through an on-board 10MΩ resistor. By default, the -15V supply is connected.

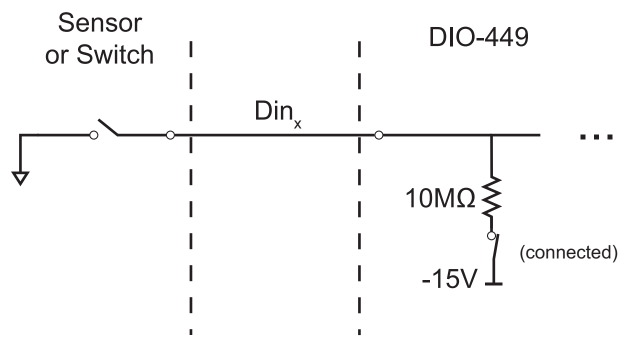


Figure B-1. GND Open Contact Monitoring with DIO-449

When the open contact switch is ON, the Din_x input is connected to GND and reads at a 0 V potential. To interpret a digital 1 on the Din_x channel, the DIO-449 high threshold value should be programmed to approximately -1.0V. The high and low threshold levels can be programmed with the low-level **DqAdv449SetLevels()** function:

```
// Set the DIO-449 board location, channels, and low and high thresholds

DqAdv449SetLevels(hd0,              //IOM handle
                  DEVN,             //Board identifier
                  CLSIZE,           //Channel list: ch(0) thru ch(CLSIZE)
                  cl,               //Array of AC/DC mode & gain settings
                  -2.0,             //(float)LEVEL_LOW threshold
                  -1.0;             //(float)LEVEL_HIGH threshold
)
```

When the open contact switch is OFF, the Din_x input is pulled down to approximately -3.3V by the -15V on-board source voltage. To interpret a digital 0 on the Din_x channel when the switch is OFF, the low threshold value should be programmed greater than -3V and less than the high threshold voltage.

For more information regarding DIO-449 programming, refer to Chapter 2 Programming with the High-Level API or Chapter 3 Programming with the Low-Level API.

B.2 VDD Open Contact Monitoring

Figure B-2 shows an example of sensor/switch output circuitry and DIO-449 input circuitry for monitoring VDD open contact switches.

In this configuration, the -15V DIO-449 on-board voltage source is connected to the Din_x input through an on-board 10MΩ resistor. By default, the -15V supply is connected; however, the -15V supply can be connected or disconnected with the low-level function, **DqAdv449SetGMux()**.

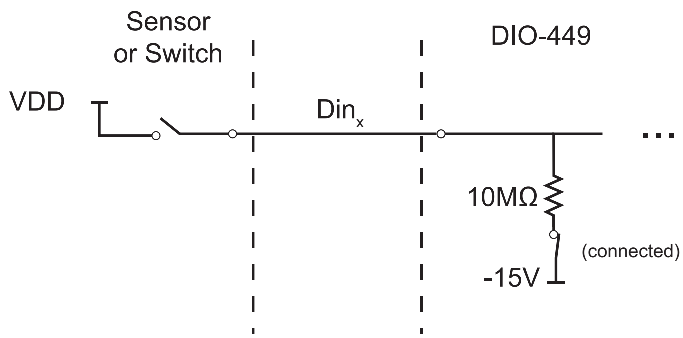


Figure B-2. VDD Open Contact Monitoring with DIO-449

When the open contact switch is ON, the Din_x input is connected to VDD. To interpret a digital 1 on the Din_x channel, the DIO-449 high threshold value should be programmed at VDD/2. The high and low threshold levels can be programmed with the low-level **DqAdv449SetLevels()** function:

```
// Set the DIO-449 board location, channels, and low and high thresholds

DqAdv449SetLevels(hd0,              //IOM handle
                  DEVN,             //Board identifier
                  CLSIZE,           //Channel list: ch(0) thru ch(CLSIZE)
                  cl,               //Array of AC/DC mode & gain settings
                  VDD/2-1.0,        //(float)LEVEL_LOW threshold
                  VDD/2;            //(float)LEVEL_HIGH threshold
)
```

When the open contact switch is OFF, the Din_x input is pulled down to approximately -3.3V by the -15V on-board source voltage. To interpret a digital 0 on the Din_x channel when the switch is OFF, the low threshold value should be programmed greater than -3V and less than VDD/2.

For more information regarding DIO-449 programming, refer to Chapter 2 Programming with the High-Level API or Chapter 3 Programming with the Low-Level API.

Index