



# **DNx-DIO-432/433 and DNx-DIO-432-800/433-800**

—

## **User Manual**

32-channel, Current Sink (DIO-432) / Current Source (DIO-433)  
Digital Output Boards  
for the PowerDNA Cube and RACK series chassis

**April 2019**

PN Man-DNx-DIO-432-433

Information furnished in this manual is believed to be accurate and reliable. However, no responsibility is assumed for its use, or for any infringement of patents or other rights of third parties that may result from its use.

All product names listed are trademarks or trade names of their respective companies.

See the UEI website for complete terms and conditions of sale:

<http://www.ueidaq.com/cms/terms-and-conditions/>



## 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](mailto:support@ueidaq.com)

Website: [www.ueidaq.com](http://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

<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Organization of this Manual	1
1.2 DIO-432/433 Board Overview	3
1.2.1 Digital Output Channels	3
1.2.2 Pulse-Width-Modulated Start / Stop / Output	3
1.2.3 Guardian Diagnostics	3
1.2.4 Guardian ADC for General Purpose Analog Input	4
1.2.5 Update Rate	5
1.2.6 Software Support	5
1.3 Features	6
1.4 Indicators	7
1.5 Specification	8
1.6 Device Architecture	10
1.6.1 PWM Functionality	10
1.6.2 Guardian ADC	10
1.7 Circuit Breakers	11
1.7.1 Configuring ADC Conversion Speed	11
1.8 Wiring & Connectors	14
1.9 PowerDNA Explorer for DIO-432/433 series	15
1.9.1 Diagnostic Readback (Input) Values	15
1.9.2 PWM Controls	16
1.9.3 Output Values	17
<b>Chapter 2 Programming with the High-level API</b>	<b>18</b>
2.1 About the High-level Framework	18
2.2 Creating a Session	18
2.3 Configuring the Resource String	18
2.4 Configuring for Digital I/O	19
2.5 Configuring the Timing	20
2.6 Configuring Pulse-width-modulation	20
2.7 Changing Circuit Breaker Limits	21
2.8 Writing Data	22
2.9 Monitoring Current and Voltage	22
2.10 Accessing Circuit Breakers	23
2.10.1 Reading Circuit Breaker Status	23
2.10.2 Resetting the Circuit Breaker	23
2.11 Cleaning-up the Session	23
<b>Chapter 3 Programming with the Low-level API</b>	<b>24</b>
3.1 About the Low-level API	24
3.2 Low-level Functions	25



3.3	Low-level Programming Techniques . . . . .	25
3.3.1	Data Transfer Modes . . . . .	26
3.4	Programming DIO-432/433 (Immediate Mode). . . . .	26
3.4.1	Resetting Circuit Breakers and Configuring Limits . . . . .	26
3.4.2	Configuring PWM Modes . . . . .	27
3.4.3	Writing Output Data . . . . .	30
3.4.4	Reading Guardian Diagnostic Values . . . . .	30
3.4.5	Reading Circuit Breaker Status & Reengaging . . . . .	31
<b>Appendix A . . . . .</b>		<b>32</b>
	Accessories . . . . .	32
<b>Appendix B . . . . .</b>		<b>33</b>
	PWM Soft Start . . . . .	33
	PWM Soft Stop . . . . .	34
	PWM Continuous Mode . . . . .	34
	Programming Notes . . . . .	34
<b>Index . . . . .</b>		<b>35</b>



# List of Figures

<b>Chapter 1 Introduction</b> .....	<b>1</b>
1-1 Schematic for Low-Side FET Digital Output in DIO-432.....	4
1-2 Schematic for High-Side FET Digital Output in DIO-433.....	4
1-3 Photo of DNA-DIO-432/433 Series Digital Output Board .....	7
1-4 DNx-DIO-432/433 series Logic Block Diagram .....	10
1-5 Sampling rate, CB react time, and noise .....	11
1-6 DIO-432 Disconnect Time for Digital Outputs.....	13
1-7 DNx-DIO-432/433 series Pinout Diagram.....	14
1-8 PowerDNA Explorer Display for Guardian Diagnostics .....	15
1-9 PowerDNA Explorer Display for PWM Controls .....	16
1-10 PowerDNA Explorer Display for Output Controls .....	17
<b>Chapter 2 Programming with the High-level API</b> .....	<b>18</b>
<b>Chapter 3 Programming with the Low-level API</b> .....	<b>24</b>
3-1 Disabling a DIO-432 Channel in Soft-both PWM Mode.....	28
<b>Appendix A</b> .....	<b>32</b>
A-1 Pinout and photo of DNA-STP-62 screw terminal panel.....	32
<b>Appendix B</b> .....	<b>33</b>
B-1 Typical Soft Start Cycle on DIO-433.....	33
<b>Index</b> .....	<b>35</b>



# Chapter 1 Introduction

This manual outlines the feature set and use of the DNx-DIO-432/433 series, 32-channel current sinking/sourcing digital output boards.

Versions supported in this manual include the following:

- DIO-432: current sinking DIO, with optional voltage and current monitoring
- DIO-433: current sourcing DIO, with optional voltage and current monitoring
- DIO-432-800: low-leakage, current sinking DIO, with optional current monitor
- DIO-433-800: low-leakage, current sourcing DIO, with optional current monitoring

**NOTE:** DIO-432 refers to both the DIO-432 and DIO-432-800 versions, and DIO-433 refers to both the DIO-433 and DIO-433-800 versions, unless otherwise noted.

The following sections are provided in this chapter:

- Organization of this Manual (Section 1.1)
- DIO-432/433 Board Overview (Section 1.2)
- Features (Section 1.3)
- Indicators (Section 1.4)
- Specification (Section 1.5)
- Device Architecture (Section 1.6)
- Circuit Breakers (Section 1.7)
- Wiring & Connectors (Section 1.8)

## 1.1 Organization of this Manual

This *DIO-432/433 Series User Manual* is organized as follows:

- **Introduction**  
 Chapter 1 provides an overview of DNx-DIO-432/433 series features, device architecture, connectivity, and logic.
- **Programming with the High-Level API**  
 Chapter 2 provides an overview of the how to create a session, configure the session, and interpret results with the Framework API.
- **Programming with the Low-Level API**  
 Chapter 3 is an overview of low-level API commands for configuring and using the DNx-DIO-432/433 series boards.
- **Appendix A - Accessories**  
 The appendix provides a list of accessories available for use with the DNx-DIO-432/433 series boards.
- **Appendix B - Soft Start/Stop/PWM**  
 This appendix describes the Soft Start/Soft Stop/Constant PWM feature for gradually increasing/decreasing and controlling average power applied to an output load.
- **Index**  
 This is an alphabetical listing of the topics covered in this manual. A glossary of terms used with the PowerDNA systems can be viewed or downloaded from [www.ueidaq.com](http://www.ueidaq.com).



## 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:



***Before plugging any I/O connector into the Cube or RACKtangle, be sure to remove power from all field wiring. Failure to do so may cause severe damage to the equipment.***

### **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.



**1.2 DIO-432/433 Board Overview**

The DNx-DIO-432/433 series are 32-channel digital output boards designed for a wide variety of digital control applications.

DNA-DIO-432/433, DNR-DIO-432/433, and DNF-DIO-432/433 board versions are compatible with the UEI Cube, RACKtangle, and FLATRACK chassis respectively. These board versions are electronically identical and differ only in 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.

**1.2.1 Digital Output Channels**

DIO-432/432-800 outputs are configured as current sinking. DIO-433/433-800 outputs are configured as current sourcing.

Each channel in the DIO-432/433 series uses an FET in the output circuit to switch voltages in the range of 3.3 to 36.0 VDC. The boards require an external 3.3 to 36 VDC power supply for the load (see **Figure 1-1** and **Figure 1-2** for diagrams).

Each channel is rated for continuous operation at 600 mA with an output voltage drop of less than 550 mV.

DIO-432/433 series boards can tolerate occasional current spikes up to 2 A per channel, but overall sink (432)/source(433) current flow should be limited to 19.2 A per I/O board. Users must ensure that their +Vcc power supply can supply enough current for the connected load.

Note that DIO-432-800 and DIO-433-800 provide higher OFF impedance and an extremely low leakage current of less than 1  $\mu$ A in the OFF state.

**1.2.2 Pulse-Width-Modulated Start / Stop / Output**

Users have the option of programming each channel to have a pulse-width-modulated (PWM) “soft-start” or “soft-stop”.

This allows power to be applied/removed gradually, greatly increasing the reliability of devices like incandescent bulbs where thermal shock reduces life expectancy.

The PWM may also be configured to run continuously as an analog output control for low speed, high voltage/current applications.

**1.2.3 Guardian Diagnostics**

The DIO-432/433 series boards are part of UEI’s Guardian diagnostic series and provide unique capabilities for monitoring outputs.

- the DIO-432/433 (non -800 versions) can monitor voltage and current
- the DIO-432-800/433-800 can monitor current

Current sense resistors mounted in the output circuits for each channel enable the boards to monitor current continuously and to detect and flag short circuits, open circuits, and other “off-normal” operating conditions for every output circuit.

The output current monitor also provides over-current protection. This feature allows users to select the trip current level and also the duration of a permitted overload, which may be set as short as 10 milliseconds. When a set point is exceeded, the circuit is shut down. Note that programming current limits is supported when channels are configured as digital outputs but is not supported in PWM mode.

Each board provides 350 Vrms isolation between I/O and chassis and any other installed I/O boards.

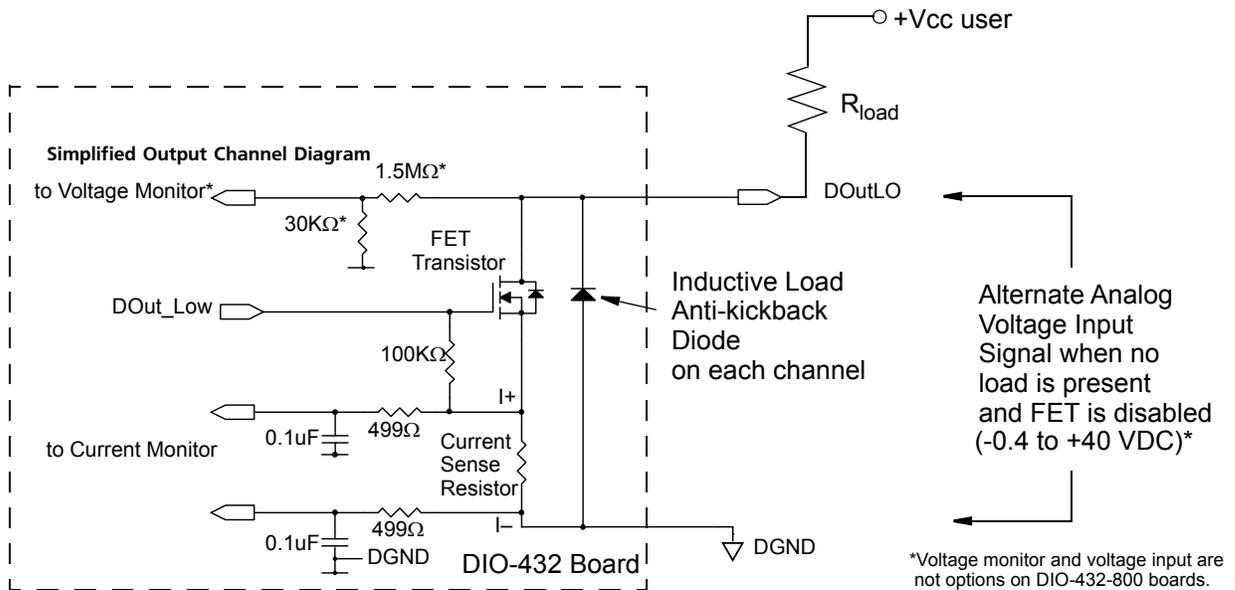


**1.2.4 Guardian ADC for General Purpose Analog Input**

For non -800 versions, any DIO-432 or DIO-433 channels that are not used as digital outputs may be used as general purpose analog voltage inputs by utilizing the voltage measurement capabilities of the Guardian ADC. This optional configuration is not available on the DIO-432-800 and DIO-433-800.

When using DIO-432 (current sinking) as a general purpose analog input:

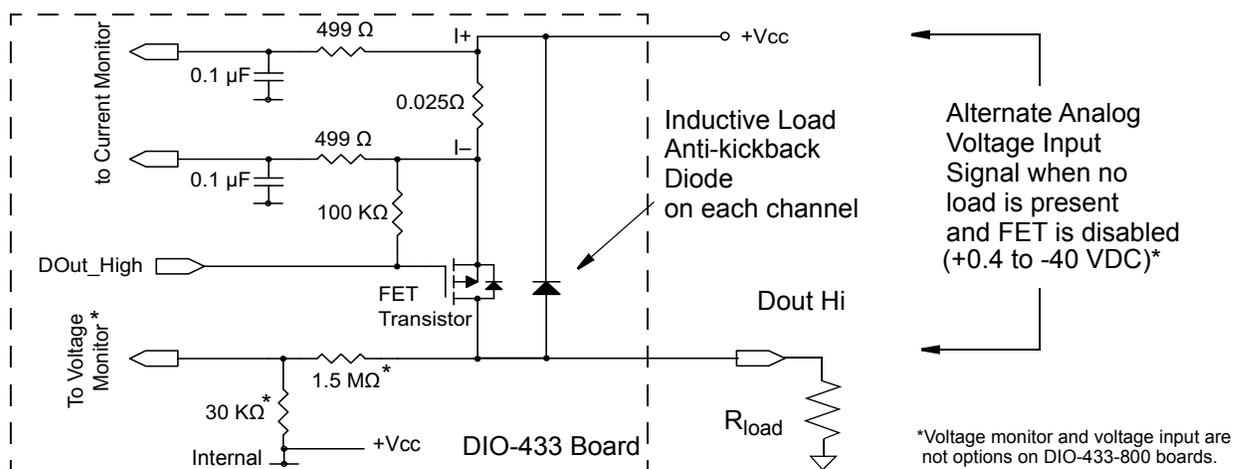
- Connect the voltage input signal between DOut LO (up to +40 VDC) and DGND (0 VDC), as shown in **Figure 1-1** below. The signal level should be in the -0.4 to +40 VDC input range relative to DGND



**Figure 1-1 Schematic for Low-Side FET Digital Output in DIO-432**

When using DIO-433 (current sourcing) as a general purpose analog input:

- Connect the voltage signal between DOut HI (up to -40 VDC) and GND (0 VDC), as shown in **Figure 1-2**. Note that the voltage input signal range can be +0.4 to -40 VDC relative to Vcc:



**Figure 1-2 Schematic for High-Side FET Digital Output in DIO-433**

**NOTE:** \*For the DIO-432-800 and DIO-433-800, the above schematics change by omitting the 1.5 MΩ / 30 kΩ resistors and Voltage Monitor.





Always keep output FETs disabled when using a channel as an analog input. As a precaution, set the circuit breaker trip current to 100  $\mu$ A through 1 mA to ensure that the measurement sensors are not damaged if a FET is accidentally turned ON.

The DIO-432 reports a positive voltage measurement between DGND and DOUT pins. The DIO-433 reports negative voltage values measured between Vcc and DOUT pins. Both boards return current (measured at the | + and | - pins of the sense resistor) as a positive number when the FET is open.

**1.2.5 Update Rate**

The I/O boards run at update rates up to 1 kHz. The states of all outputs are updated at once by transferring a single, 32-bit word via UEI write API.

**1.2.6 Software Support**

Software included with the DNx-DIO-432/433 series provides a comprehensive yet easy to use API that supports Windows, Linux, QNX, VxWorks, and most other popular real-time operating systems. Windows users may take advantage of the UEIDAQ Framework, which provides a simple and complete software interface to all popular Windows programming language and data acquisition and control applications, such as LabVIEW and MATLAB.



## 1.3 Features

The common features of the DNx-DIO-432/433 series are listed below:

- 32 digital output channels:
  - 32 current sourcing (DNx-DIO-433, DNx-DIO-433-800)
  - 32 current sinking channels (DNx-DIO-432, DNx-DIO-432-800)
- Over-current protection user-programmable from 50 mA to 2 A and stored in EEPROM
- Over-current response time user-programmable from 10 to 5000 ms
- Output throughput rate of 1000 samples per second (max)
- 600 mA per channel of continuous output current, 2 A momentary
- Maximum total current 19.2 A aggregate for all channels (per I/O board)
- Built-in diagnostic monitoring on each output channel provides fault detection and simplifies system diagnostics (1% or better accuracy at  $\pm 50$  V input range)
- Built-in current and voltage monitoring on DIO-432/433 output channels; Built-in current monitoring on DIO-432-800/433-800 output channels
- Wide 3.3 V to 36 V operating range (requires external power supply for load)
- Inductive load kickback protection diodes on every channel
- Intelligent electronic current monitoring ( $\pm 0.5\%$  of F.S. accuracy) on every channel with circuit breaking via FET disable
- Power consumption of the board is 2.5 W with all outputs driving 600 mA
- Over-current per channel limits
- Auto-enable option is selectable per channel. It will try to restore disabled channel functionality after a programmable delay that can be up to 65 seconds long (settable in 15 nS increments)
- Guaranteed output disabled (OFF) state under the following conditions: initial power-up, external power OFF, overload detected
- PWM (slow-ON/slow-OFF or constant PWM) function on every output

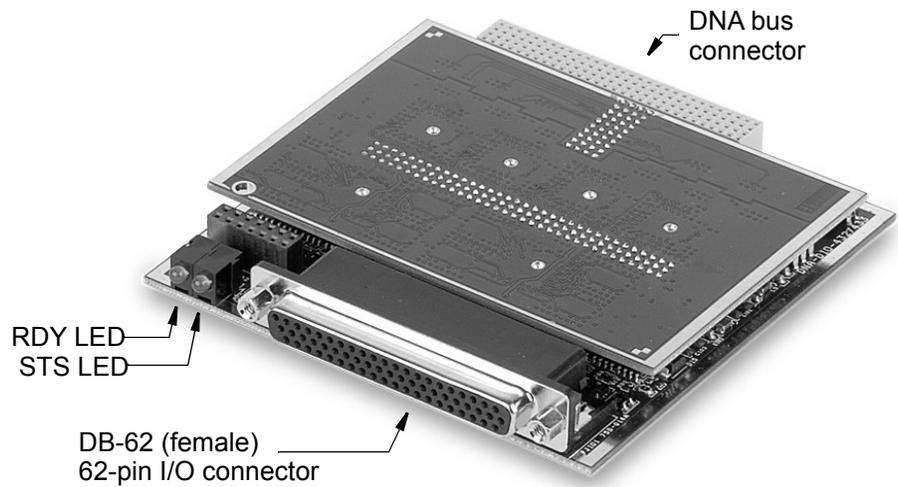


## 1.4 Indicators

The DNx-DIO-432/433 series indicators are described in **Table 1-1** and illustrated in **Figure 1-3**.

**Table 1-1 DIO-432/433 series Indicators**

LED Name	Description
RDY	Indicates board is powered up and operational
STS	Indicates which mode the board is running in: <ul style="list-style-type: none"> <li>• <b>OFF</b>: Configuration mode, (e.g., configuring channels, running in point-by-point mode)</li> <li>• <b>ON</b>: Operation mode</li> </ul>



**Figure 1-3 Photo of DNA-DIO-432/433 Series Digital Output Board**

**1.5 Specification** The technical specifications for the DIO-432/433 series are provided in the tables below:

**Table 1-2. DNx-DIO-432/433 Technical Specifications**

Number of channels	32 digital outputs
Output configuration	Current sink (DIO-432) / source (DIO-433)
Output port configuration	Single 32-bit word
Output Drive	600 mA per channel continuous; 3.5 A peak (100 mS max)
Output ON voltage	< 550 mV @ 600 mA (incl. std 3' cable)
Output ON impedance	< 0.9 Ohm (including std 3' cable)
Output OFF impedance	> 1 Meg Ohm
Output OFF leakage	< 25 $\mu$ A
Overvoltage protection	$\pm$ 40 VDC (reverse current must be limited to 1 A to prevent damage)
Overcurrent protection Current Limit Overload response time	50 mA - 2 A 10-5000 ms (user programmable)
Output Monitoring Configuration Voltage Accuracy Current Accuracy	Multiplexed $\pm$ 10 mV max (sampled at 2 Hz) $\pm$ 1 mA, max (sampled at 10 Hz)
Soft-Start/Stop duration	256 $\mu$ S to 5 seconds
Steady State PWM output	0 to 100% in 0.4% increments. (Minimum period is 256 $\mu$ Sec)
Output Throughput Rate	1 kHz max
Power up / reboot state	Off
Power dissipation	< 2 W, not including output switches
Isolation	350 Vrms
Operating Temp. Range	Tested -40 to +85 $^{\circ}$ 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	260,000 hours

The technical specifications for the DNx-DIO-433 I/O boards are the same as those for the DIO-432 except for the following items:

Specifacaton Item	DIO-432	DIO-433
Output Configuration	Current Sink	Current Source
Output ON Voltage Drop	< 550 mV at rated 600 mA	< 500 mV at rated 600 mA



**Table 1-3. DNx-DIO-432-800 Technical Specifications**

Number of channels	32 digital outputs
Output configuration	Current sink
Output port configuration	Single 32-bit word
Output Drive	600 mA per channel continuous; 3.5 A peak (100 mS max)
Output ON voltage	< 550 mV @ 600 mA (incl. std 3' cable)
Output ON impedance	< 0.9 Ohm (including std 3' cable)
Output OFF impedance	> 30 Meg Ohm
Output OFF leakage	< 1 $\mu$ A
Overvoltage protection	$\pm$ 40 VDC (reverse current must be limited to 500 mA to prevent damage)
Overcurrent protection	
Current Limit	50 mA - 2 A
Overload response time	10-5000 ms (user programmable)
Output Monitoring	
Configuration	Multiplexed
Current Accuracy	$\pm$ 1 mA, max (sampled at 10 Hz)
Soft-Start/Stop duration	256 $\mu$ S to 5 seconds
Steady State PWM output	0 to 100% in 0.4% increments
Output Throughput Rate	1 kHz max
Power up / reboot state	Off
Power dissipation	< 2 W, not including output switches
Isolation	350 Vrms
Operating Temp. Range	Tested -40 to +85 $^{\circ}$ 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	260,000 hours

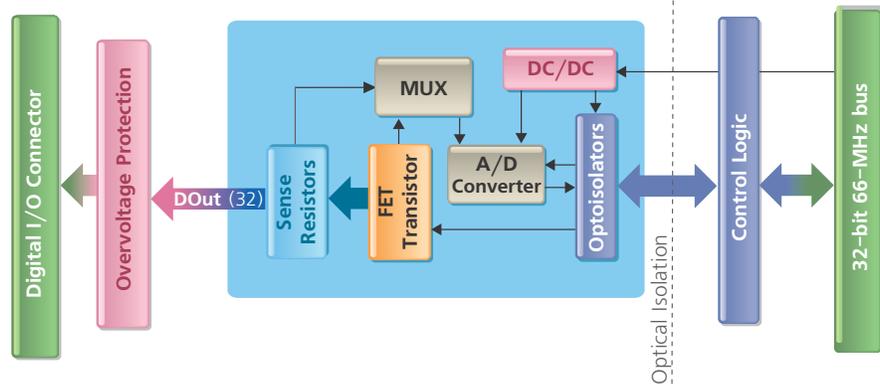
**Table 1-4. DNx-DIO-433-800 Technical Specifications**

Number of channels	32 digital outputs
Output configuration	Current source
Output port configuration	Single 32-bit word
Output Drive	600 mA per channel continuous; 3.5 A peak 10% Duty Cycle 100 ms max)
Output ON voltage drop	< 500 mV @ 600 mA (Incl. std 3' cable)
Output ON impedance	< 0.9 Ohm (Includes standard 3' cable)
Output OFF impedance	> 30 Meg Ohm
Output OFF leakage	< 1 $\mu$ A
Overvoltage protection	$\pm$ 40 VDC (reverse current must be limited to 1 A to prevent damage)
Overcurrent protection	
Current Limit	50 mA - 2 A (user-programmable)
Overload response time	10-5000 ms (user-programmable)
Output Monitoring	
Configuration	Multiplexed, one A/D
Current Accuracy	$\pm$ 1 mA, max (sampled at 10 Hz)
Soft-Start/Stop Duration	256 $\mu$ S to 5 seconds
Steady State PWM Output	0 to 100% in 0.4% increments (Minimum period is 256 $\mu$ S)
Output Throughput Rate	1 kHz max
Power up / reboot state	Off
Isolation	350 Vrms
Power dissipation	< 2 W, not including output switches
Operating Temp. Range	Tested -40 to +85 $^{\circ}$ 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	260,000 hours



## 1.6 Device Architecture

A block diagram of a DIO-432/433 series I/O board is shown below.



**Figure 1-4 DNx-DIO-432/433 series Logic Block Diagram**

Board logic is divided into isolated and non-isolated sections (**Figure 1-4**).

The isolated side handles all functions associated with switching the output loads, sensing current and voltage, and providing over-current and voltage monitoring and control. It also provides anti-kickback protection for inductive loads.

The DIO-432/433 series boards are two-board units, comprised of a layer-specific base board (made in DNA, DNR, and DNF versions) and a plug-in layer-specific ADC board. The base boards contain the FETs, sensing resistors, DC/DC module, and CLI logic; the plug-in boards contain the ADCs, buffers, and associated circuits.

A DC/DC module supplies 5 V for controlling the FETs and powering the ADC converters. On initial power-up, the DC/DC is disabled. While the DC/DC is disabled, all output FETs are in disconnect mode.

### 1.6.1 PWM Functionality

The DIO-432/433 series boards can be configured to provide slow-ON, slow-OFF, or constant PWM function on every output. Refer to Appendix B for descriptions of the Soft Start/Soft Stop/Constant PWM features.

Note that programming current limits is supported when channels are configured as current sinking or sourcing digital outputs but is not supported in PWM mode.

### 1.6.2 Guardian ADC

Diagnostic current monitoring is accomplished with per-channel 24-bit  $\Sigma\Delta$  analog-to-digital converters (Guardian ADCs). Currents up to a maximum of  $\pm 2$  A can be detected by the board. The DIO-433 and DIO-432 (non -800 versions) can additionally monitor voltage.

## 1.7 Circuit Breakers

Each output channel is equipped with a FET that can act as a circuit breaker. Refer to **Figure 1-1** and **Figure 1-2** for location.

Voltages from current sensing resistors are read by the 24-bit Guardian ADCs. The ADC outputs are then processed in logic to perform a virtual circuit breaker function. The outputs are first compared to preset limits. If measured values exceed the limits, the FETs are turned OFF, and the output circuit is shut down. Depending on how the channel is configured, the shutdown may either be immediate or delayed by a programmable time or by a cumulative number of detected faults. The current limit is up to +2 A.

Note that programming a current limit is supported when channels are configured as current sinking or sourcing digital outputs but is not supported in PWM mode.

The circuit breaker function can also be configured for either User Re-enable (default) or for Auto Re-enable. The user-re-enable mode requires a write operation to re-enable output on the disabled channel. The auto-re-enable mode attempts to re-enable the channel after a 1 second (default) delay. If an over-current is detected on restart, the channel is disabled again and the re-enable attempt is repeated. The re-enable time interval is user programmable.

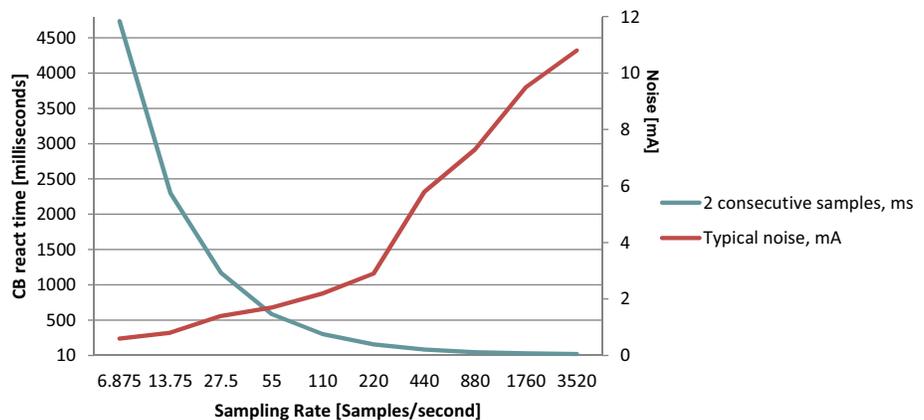
### 1.7.1 Configuring ADC Conversion Speed

The speed of the Guardian ADC is user-programmable in the range from 0.6 to 293 Hz. Refer to `CreateDOProtectedChannel()` when using Framework API (**Chapter 2**). Refer to `DqAdv432SetAll()` when using low-level API (**Chapter 3**).

The default ADC speed is 13 Hz per channel. Slower speed produces more accurate results but increases circuit breaker disconnect time.

**Table 1-5** provides the ADC speeds, the time required to react to over-current and shut down the output circuit using the circuit breakers, and the accuracy.

Note that accuracy is best at low sampling rates, and the reaction time is quickest at high sampling rates.



**Figure 1-5 Sampling rate, CB react time, and noise**

As a guideline, you should choose a sampling rate that is low enough to meet your accuracy need, but fast enough to allow the circuit breaker the time needed to disconnect in case of over-current.



Note that the over-current limit is software configurable when the channel is programmed as a current sinking or sourcing digital output (not in PWM mode; Current limits for circuit breakers in PWM mode are not applicable).

Set sampling rate and limits to avoid over-current for a prolonged period; failing to do so may permanently damage your external equipment and/or the I/O board.

**NOTE:** For detailed instructions for configuring the DIO-432/433 series, refer to the Framework API, described in **Chapter 2**, and/or the low-level API, described in **Chapter 3**.

The table below provides sample rate, typical circuit breaker disconnect time in milliseconds, and noise in milliamps when channels are configured as digital outputs.

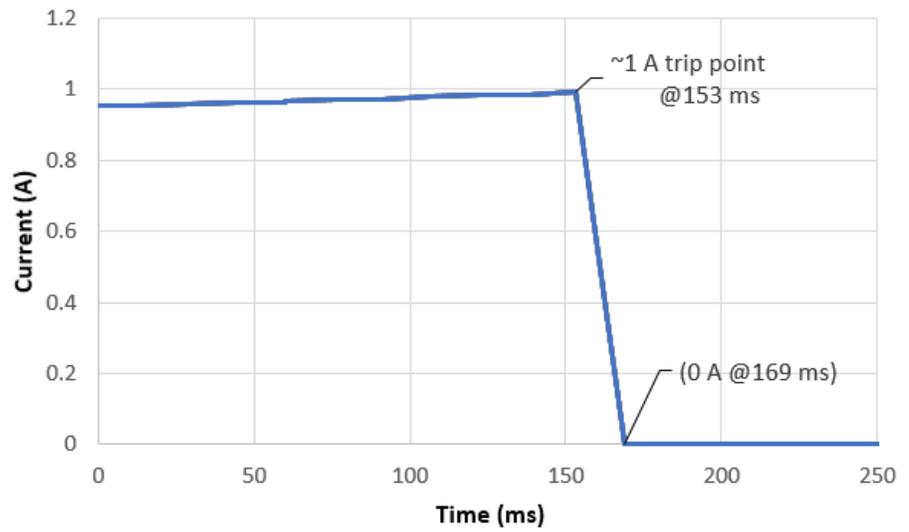
**Table 1-5. Sample rate, disconnect time, and accuracy**

Sample Rate (S/s)	Circuit Breaker Disconnect Time, ms				Typical noise, mV, Vin	Typical noise, mA
	Immediate decision, ms	2 consecutive samples, ms	3 consecutive samples	4 consecutive samples		
3520	10	20	30	41	46	10.8
1760	15	29	44	58	43	9.5
880	24	47	71	94	31	7.3
440	42	83	125	166	25	5.8
220	78	156	233	311	11	2.9
110	150	301	451	602	9.2	2.2
55	293	586	878	1171	7.1	1.7
27.5	586	1171	1757	2342	6.5	1.4
13.75	1152	2304	3456	4608	3.1	0.8
6.875	2368	4736	7104	9472	2.4	0.6

By default, the DNx-DIO-432 digital output circuitry is configured to trip the circuit breaker at 2 A in less than 20 ms.



The graph below shows breaker disconnect time (< 20 ms) when the limit is set to 1 A:

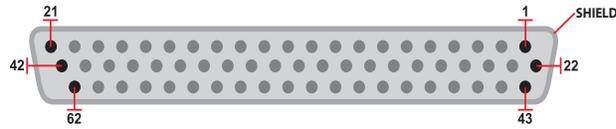


**Figure 1-6 DIO-432 Disconnect Time for Digital Outputs**



## 1.8 Wiring & Connectors

The pinout for the DIO-432/433 series is provided below:



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

Note: For rated performance all ground pins should be connected to the external ground.

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

Note:  
For rated performance all +Vcc pins should be connected to +Vcc.

Note: Do NOT make any connections to pins marked "Rsvd".

Figure 1-7 DNx-DIO-432/433 series Pinout Diagram



**Before plugging any I/O connector into the chassis or I/O board, be sure to remove power from all field wiring. Failure to do so may cause severe damage to the equipment.**

### 1.9 PowerDNA Explorer for DIO-432/433 series

PowerDNA Explorer is a GUI-based application for communicating with your RACK or Cube system. You can use it to start exploring a system and individual boards in the system. PowerDNA Explorer is provided in the installation directory.

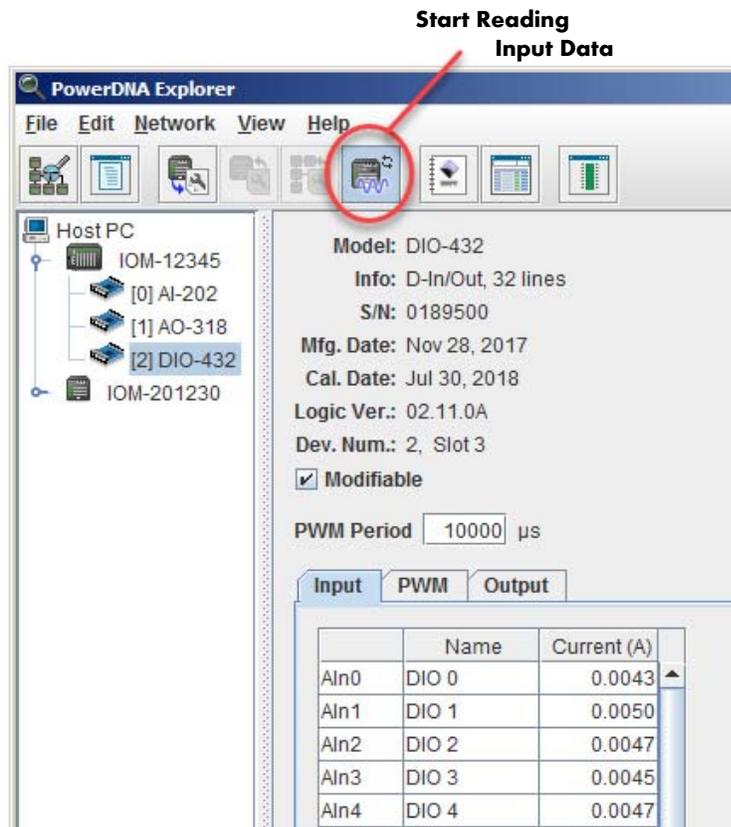
When using PowerDNA Explorer to explore a DIO-432/433 series board, note that the right-hand panel contains three tabs:

- **Input:** displays diagnostic readback current
- **PWM:** displays pulse-width-modulation controls
- **Output:** sets ON/OFF output states immediately; initialization output state applied at power-up; and output state applied at shutdown

#### 1.9.1 Diagnostic Readback (Input) Values

The PowerDNA Explorer **Input** tab for the DIO-432/433 series provides diagnostic current readings:

- **Alnx:** the DIO channel number.
- **Name:** a name or note that you wish to give to the channel.
- **Current (A):** diagnostic reading of the current (A).



**Figure 1-8 PowerDNA Explorer Display for Guardian Diagnostics**

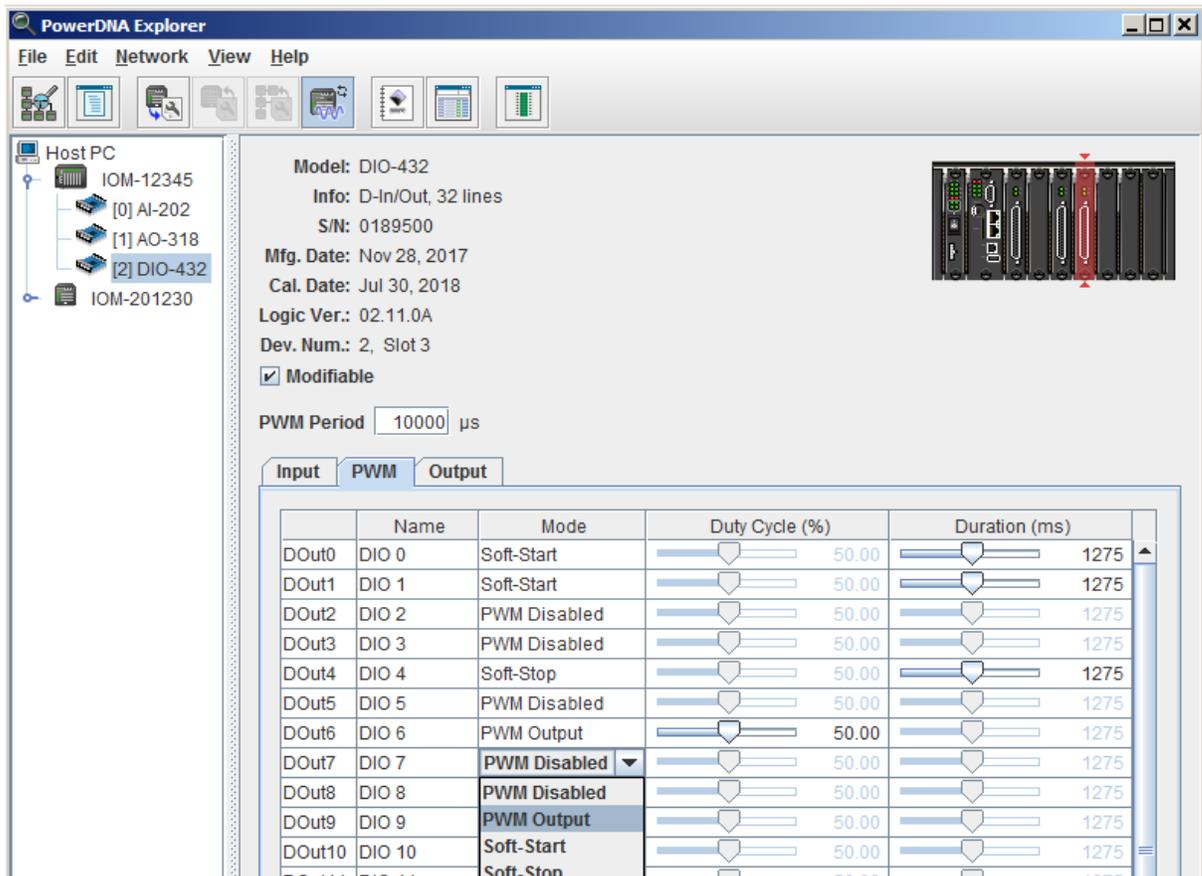
To read diagnostic values, click the **Start Reading Input Data** button.

**NOTE:** If current values are not updating, verify the **Modifiable** checkbox above the tabs is checked.



**1.9.2 PWM Controls** The PowerDNA Explorer **PWM** tab for the DIO-432/433 series contains the following columns:

- **DOutx:** the channel number.
- **Name:** a name or note that you wish to give to the channel.
- **Mode:** PWM mode to apply to channel:
  - **PWM Disabled:** do not modulate the output.
  - **Soft-Start:** upon enabling the DOut channel via the **Output** tab, modulate the output for the number of milliseconds specified in the **Duration** column.
  - **Soft-Stop:** upon disabling the DOut channel via the **Output** tab, modulate the output for the number of milliseconds specified in the **Duration** column.
  - **PWM Output:** use the channel as a continuously modulated output (**Output** tab will not affect output state). Regulate high/low duty cycle via the **Duty Cycle (%)** column, where the duty cycle period is programmed via the **PWM Period** textbox.
- **Duty Cycle (%):** controls the percent low vs high time of the output when in the **PWM Output** mode. To change the duty cycle values, verify **Modifiable** is checked
- **Duration (ms):** controls how long outputs will be modulated upon enabling DOut in soft-start mode or disabling in soft-stop mode.



**Figure 1-9 PowerDNA Explorer Display for PWM Controls**

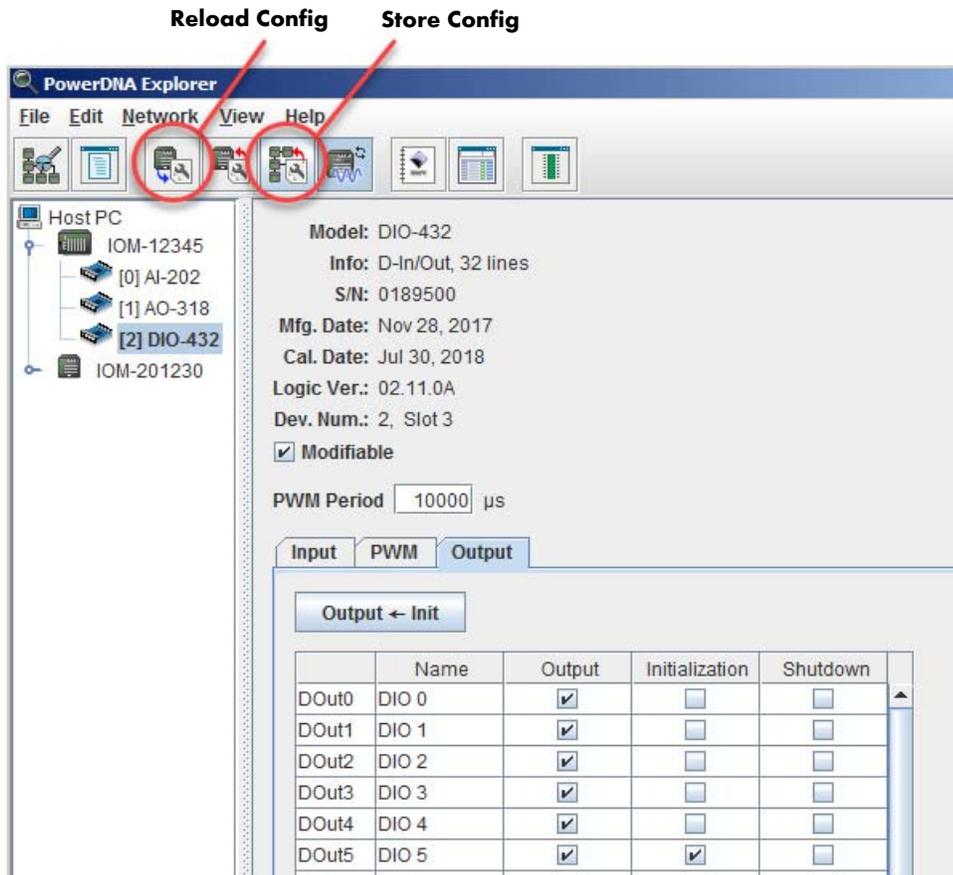


**1.9.3 Output Values** The PowerDNA Explorer **Output** tab for the DIO-432/433 series contains the following columns:

- **DOutx**: the channel number.
- **Name**: a name or note that you wish to give to the channel.
- **Output**: controls for immediately enabling or disabling of the corresponding digital output channel.
- **Initialization**: stores values into the EEPROM configuration to set the output state on power-up. Factory default is OFF.
- **Shutdown**: stores values into the EEPROM configuration to set the output state on shutdown. Factory default is OFF.

Changes to the output states using this display are written instantaneously to hardware.

To change the initialization or shutdown states, verify **Modifiable** is checked, and then enter new values. After updating fields, press the **Store Config** button to update the value in hardware. You can alternatively load a previously stored configuration with the **Reload Config**.



**Figure 1-10 PowerDNA Explorer Display for Output Controls**



## Chapter 2 Programming with the High-level API

This chapter provides the following information about using the UeiDaq high-level Framework API to program the DNx-DIO-432/433:

- About the High-level Framework (Section 2.1)
- Creating a Session (Section 2.2)
- Configuring the Resource String (Section 2.3)
- Configuring for Digital I/O (Section 2.4)
- Configuring the Timing (Section 2.5)
- Configuring Pulse-width-modulation (Section 2.6)
- Changing Circuit Breaker Limits (Section 2.7)
- Writing Data (Section 2.8)
- Monitoring Current and Voltage (Section 2.9)
- Accessing Circuit Breakers (Section 2.10)
- Cleaning-up the Session (Section 2.11)

### 2.1 About the High-level Framework

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.

UeiDaq Framework is bundled with examples for supported programming languages. Examples are located under the UEI programs group in:

- *Start » Programs » UEI » Framework » Examples*

The following sections focus on the C++ API, but the concept is the same no matter which programming language you use.

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

### 2.2 Creating a Session

The Session object controls all operations on your PowerDNx device. Therefore, the first task is to create a session object:

```
// create a session object
CUEISession session;
```

**NOTE:** All tripped breakers will be reset to default states when starting a session.

### 2.3 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 output port 0 on device 1 at IP address 192.168.100.2: "pdna://192.168.100.2/Dev1/Do0"



## 2.4 Configuring for Digital I/O

You use a Protected Output session to configure DIO-432/433 channels as digital outputs with Guardian features, including circuit breaker protection.

**NOTE:** In Framework, a digital channel corresponds to a physical port on the device. You cannot configure a session only to access a subset of lines within a digital port.

**NOTE:** Sessions are unidirectional. The DIO-432/433 is also unidirectional and you only need to configure one session for output.

The DIO-432/433 is an intelligent digital output device that can monitor the current flowing through each of its digital lines and open a line when the current exceeds specified current limits.

You can configure the device so that when an over current condition occurs, it will close the connection after a programmed delay (if the current returns within the limits).

The device is also capable of generating a pulse train for each rising and/or falling edge, thus allowing for modulated (soft) start and/or stop; however, in PWM modes current limits are not applicable.

The following call configures the digital output port of a DNx-DIO-432/433 set as device 1. Note that port Do0 is used to access all 32 channels.

```
// Configure session to write to port 0 on device 1
session.CreateDOProtectedChannel("pdna://192.168.100.2/Dev1/Do0",
    -0.01,
    0.01,
    200.0,
    false,
    50.0);
```

CreateDOProtectedChannel configures the following parameters:

- **underCurrentLimit:** not supported for this product; program to negative <overCurrentLimit>
- **overCurrentLimit:** when the current goes above this limit, the line opens (double in amperes).
- **currentSamplingRate:** the rate at which the DNx-DIO-432/433 monitors current (double in Hz).  
This rate has a direct influence on how fast the DNx-DIO-432/433 reacts to an over-current condition. Refer to Section 1.7.1 on page 11 for more information.
- **autoRetry:** specifies whether the DNx-DIO-432/433 attempts to close the circuit after an over current condition (boolean).
- **retryRate:** specifies how often the DNx-DIO-432/433 attempts to close the circuit (double).



## 2.5 Configuring the Timing

You can configure the DIO-432/433 to run in simple mode (point by point).

In simple mode, the delay between samples is determined by software on the host computer.

The following sample shows how to configure the simple mode. Please refer to the *UeiDaq Framework User Manual* to learn how to use other timing modes.

```
// configure timing for point-by-point (simple mode)
session.ConfigureTimingForSimpleIO();
```

## 2.6 Configuring Pulse-width-modulation

PWM features are programmable on a per-output line basis. You can use methods available in the `CUeiDOProtectedChannel` class to program the PWM mode, period, duty cycle and/or duration for specified output lines.

First, get a pointer to the channel object:

```
CUeiDOProtectedChannel* pChan =
    dynamic_cast<CUeiDOProtectedChannel*>(session.GetChannel(index));
```

All lines will run at the same frequency. The following code programs the pulse period to 50  $\mu$ s:

```
pChan->SetPWMPeriod(50);
```

You can configure any of the 32 output lines to generate a pulse train continuously or on a low-to-high or high-to-low transition:

- `UeiDOPWMDisabled`: Disable PWM transitions on a channel
- `UeiDOPWMSoftStart`: Use a pulse train for low-to-high transitions
- `UeiDOPWMSoftStop`: Use a pulse train for high-to-low transition
- `UeiDOPWMSoftBoth`: Use a pulse train for both transitions
- `UeiDOPWMContinuous`: Continuously output a PWM pulse train

The following code configures output line 0 to generate pulses continuously and line 12 to generate pulses on low-to-high and high-to-low transitions:

```
pChan->SetPWMMode(0, UeiDOPWMContinuous);
pChan->SetPWMMode(12, UeiDOPWMSoftBoth);
```

We must now specify the duty cycle of the lines configured in continuous mode:

```
pChan->SetPWMDutyCycle(0, 0.5);
```



Also specify the length in  $\mu\text{s}$  of the pulse train generated on lines configured for soft start/stop mode:

```
pChan->SetPWMLength(12, 1000);
```

**NOTE:** When a line is configured to generate pulses continuously, write operations (as described in Section 2.8 below) are ignored. The line will keep generating pulses no matter what value is written to it.

A line configured for soft start will generate a pulse train after writing 1 if its previous state was 0.

A line configured for soft stop will generate a pulse train after writing 0 if its previous state was 1.

## 2.7 Changing Circuit Breaker Limits

The circuit breaker for a digital output trips when the corresponding channel current reads outside of the programmable limits. Current limits for all digital outputs are initialized to the same value when creating the session.

**NOTE:** Current limits apply to channels when configured as a current sourcing / sinking digital outputs, not in PWM mode. Current limits for circuit breakers in PWM mode are not applicable.

Before starting your session, you can change limits per digital output channel using the `SetOverCurrentLimit()` methods available in the `CUeiDOProtectedChannel` class.

The following example changes the over current limit to 1.0 amp on digital output line 10:

```
// first create a pointer to the output channel (if you haven't
// already) to program settings

CUeiDOProtectedChannel* pChan =
    dynamic_cast<CUeiDOProtectedChannel *>(session.GetChannel(index));

// change channel 10 over current limit to 1A

pChan->SetOverCurrentLimit(10, 1.0);
```



## 2.8 Writing Data

Writing data is done using a writer object.

The following sample shows how to create a writer object and write data:

```
// create a writer and link it to the session's stream
CUEiDigitalWriter writer(session.GetDataStream());

// to write a value, the buffer must contain one value per channel
uint32 data = 0xFEFE;

// write one scan, the buffer must contain one value per channel
writer.WriteSingleScan(&data);
```

## 2.9 Monitoring Current and Voltage

You can monitor the current and voltage at each digital line.

Use an Analog Input session the same way you would measure voltage from an Analog Input device.

You can read up to 64 diagnostic channels:

- The first 32 channels (0 to 31) read the current flowing through each of the output lines.
- The next 32 channels (32 to 63) read the voltage at each output line.

**NOTE:** For the DIO-432-800 and DIO-433-800 board versions, you can only monitor current. Monitoring voltage on the -800 versions is not available.

The following code shows how to measure current out of the first 4 digital lines:

```
// create & configure session
CUEiSession aiSs;
aiSs.CreateAIChannel("pdna://192.168.100.2/Dev1/Ai0:3"
                    -10.0, 10.0,
                    UeiAIChannelInputModeDifferential);
aiSs.ConfigureTimingForSimpleIO();

// create & configure reader; create buffer
CUEiAnalogScaledReader aiReader(aiSs.GetDataStream());
double currents[8];

// read monitored current into buffer
aiReader.ReadSingleScan(currents);
```



**2.10 Accessing Circuit Breakers**

The DIO-432/433 circuit breakers can be accessed with a `CUeiCircuitBreaker` object.

The first step in accessing the circuit breakers is to create the object:

```
// Create a circuit breaker and link it to the session's stream
CUeiCircuitBreaker cb(session.GetDataStream(), 0);
```

**2.10.1 Reading Circuit Breaker Status**

Call the `ReadStatus` method to retrieve the DIO-432/433 CB status masks.

The circuit breakers will trip when the default minimum or maximum conditions are exceeded. Minimum and maximum limits are initialized for the board in the `CreateDOProtectedChannel()` call when creating the session and can be customized per digital output channel (see Section 2.7).

Each bit in the `current` status mask corresponds to a circuit breaker: 1 if the CB on `Dout0` is currently tripped, 0 otherwise; 2 if the CB on `Dout1` is currently tripped, etc.

Each bit in the `sticky` status mask corresponds to a circuit breaker: 1 if CB was tripped at least once since last time status was read, 0 otherwise, etc.

```
// Read CB status
uint32 currStatus, stickyStatus;
cb.ReadStatus(&currStatus, &stickyStatus);
```

**2.10.2 Resetting the Circuit Breaker**

Use the `Reset` method on a circuit breaker object to reset one or more breakers.

The mask parameter specifies which circuit breaker to reset (1 to reset, 0 to leave alone).

```
// Reset breakers for digital output channels 0 and 2
cb.Reset( 1<<0 | 1<<2 );
```

**2.11 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();
```



## Chapter 3 Programming with the Low-level API

This chapter provides the following information about programming the DIO-432/433 using the low-level API:

- About the Low-level API (Section 3.1)
- Low-level Functions (Section 3.2)
- Low-level Programming Techniques (Section 3.3)
  - Data Transfer Modes (Section 3.3.1)
- Programming DIO-432/433 (Immediate Mode) (Section 3.4)
  - Resetting Circuit Breakers and Configuring Limits (Section 3.4.1)
  - Configuring PWM Modes (Section 3.4.2)
  - Writing Output Data (Section 3.4.3)
  - Reading Guardian Diagnostic Values (Section 3.4.4)
  - Reading Circuit Breaker Status & Reengaging (Section 3.4.5)

### 3.1 About the Low-level API

The low-level API provides direct access to the DAQBIOS protocol structure and registers in C. The low-level API is intended for speed-optimization, when programming unconventional functionality, or when programming under Linux or real-time operating systems.

When programming in Windows OS, however, we recommend that you use the UeiDaq high-level Framework API (see **Chapter 2**). The Framework extends the low-level API with additional functionality that makes programming easier and faster, and additionally the Framework supports a variety of programming languages and the use of scientific software packages such as LabVIEW and MATLAB.

For additional information regarding low-level programming, refer to the *PowerDNA API Reference Manual* located in the following directories:

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



### 3.2 Low-level Functions

Table 3-1 provides a summary of DIO-432/433-specific functions. All low-level functions are described in detail in the *PowerDNA API Reference Manual*.

**Table 3-1 Summary of Low-level API Functions for DNx-DIO-432/433**

Function	Description
DqAdv432Reengage	Resets circuit breakers.
DqAdv432SetLimit	Sets the over-current limits for a channel (not in PWM mode).
DqAdv432SetPWM	Configures the PWM mode for a channel.
DqAdv40xWrite	Writes the output state for each of the digital outputs.
DqAdv432GetAll	Gets two structures with Guardian values, circuit breaker states, and calibrated data.
DqAdv432SetAll	Sets most of the configuration parameters.
DqAdv40xRead	Reads the trip/not-tripped status of each circuit breaker.

### 3.3 Low-level Programming Techniques

Application developers are encouraged to explore existing source code examples when first programming the DIO-432/433. 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*

Sample code has the name of the I/O board(s) being programmed embedded in the sample name. For example, `SampleRTDMap43x` contains sample code for running an DIO-432/433 in RTDMap mode. Refer to the low-level sample code `Sample432_433` for an example of how to use the Guardian diagnostic features.



### 3.3.1 Data Transfer Modes

The DIO-432/433 supports the following acquisition modes.

- **Immediate (point-to-point):** Designed to provide easy access to a single I/O board at a non-deterministic pace. Runs at a maximum of 100 Hz.
- **RTDMap:** Designed for closed-loop (control) applications. Users set up a “map” of I/O boards and channels from which to read or write incoming or outgoing data. A single API call (refresh) paced by the user application causes data to be transferred directly between I/O board FIFOs and the user application.
  - RTDMap delivers 1 data sample per channel per transfer

**NOTE:** API that implement data acquisition modes and additional mode descriptions are provided in the *PowerDNA API Reference Manual*.

### 3.4 Programming DIO-432/433 (Immediate Mode)

The following sections provide an overview of how to set up and use your DIO-432/433 in Immediate Mode using the low-level API.

For best results, use this overview in conjunction with actual sample code, (e.g., `Sample432_433`). Refer to Section 3.3 for sample code location.

This overview does not address opening communication with the RACK or Cube, handling error codes, and other tasks not unique to DIO-432/433 configuration and usage (examples of these tasks can be found in the DIO-432/433 sample code).

#### 3.4.1 Resetting Circuit Breakers and Configuring Limits

At the start of your application, use the `DqAdv432Reengage()` API to reset circuit breakers and the `DqAdv432SetLimit()` API to configure over-range limits.

**NOTE:** Current limits apply to channels when configured as a current sourcing / sinking digital outputs, not in PWM mode. Current limits for circuit breakers in PWM mode are not applicable.

To reset circuit breakers, provide `DqAdv432Reengage()` a bitwise mask of which circuit breakers to reset: `0x00000001` resets the circuit breaker on DO0, `0x00000002` resets the circuit breaker on DO1, `0x00000004` resets the circuit breaker on DO2, etc.

The following resets circuit breakers on all 32 digital output channels:

```
// reset all 32 circuit breakers
DqAdv432Reengage(hd, DEVN, 0xFFFFFFFF);
```



Pass `DqAdv432SetLimit()` a channel identifier representing the digital output channel to set limits for. Limits are programmed in amperes.

- To set over-current limits, identify the channel as 0..31 to correspond to digital output channel 0 through 31.

The following example sets the over-current limit to 1.0 A for all channels:

```
// set current limits;
// CHANNELS is a #define constant set to 32

for (i = 0; i < CHANNELS; i++) {
    DqAdv432SetLimit(hd, DEVN, i, 1.0);
}
```

### 3.4.2 Configuring PWM Modes

You have the option of configuring a digital channel to use pulse-width-modulator (PWM) features, which are programmable on a per-output line basis.

To configure these options, use the `DqAdv432SetPWM()` API to set the PWM period, mode, and duty cycle for the channel(s).

PWM functionality includes modes to disable or to generate a pulse train continuously or on a low-to-high and/or high-to-low transition:

- `DQDIO432_PWM_DISABLED`: Disable PWM transitions on a channel
- `DQDIO432_PWM_SOFTSTART`: Use a pulse train for low-to-high transitions
- `DQDIO432_PWM_SOFTSTOP`: Use a pulse train for high-to-low transition
- `DQDIO432_PWM_SOFTBOTH`: Use a pulse train for both transitions
- `DQDIO432_PWM_MODE`: Continuously output a PWM pulse train: the channel output cannot be turned on and off with writes of 1 and 0 to the channel control bit.
- `DQDIO432_PWM_MODE_GATED`: PWM pulse train output while channel is ON (written with a 1) and OFF when written with a 0.

The following code configures output channel 2 to generate modulated pulses on low-to-high and high-to-low transitions (soft-both) for a length of time equal to 16 PWM periods:

```
// set the period of a single pulse period in microseconds
// programmed for all 32 dout channels (not per-channel programmable)

period_us = 10000; // PWM period

// set channel, mode and duty cycle (only 1 channel in data_pwm array)
```



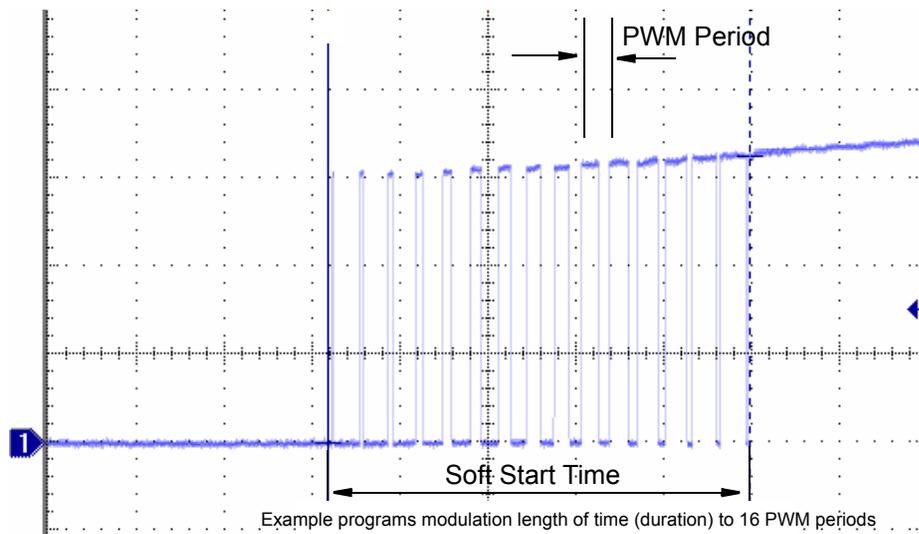
```

// set the period of a single pulse period in microseconds
// programmed for all 32 dout channels (not per-channel programmable)

data_pwm[0].mode = DQDIO432_PWM_SOFTBOTH;
data_pwm[0].channel = 2; // set to DOut2 channel
data_pwm[0].duty_cycle_length = 160000; // 16*period_us (Figure 3-1)

// set the period of the pulse width modulation in microseconds

DqAdv432SetPWM(hd, // handle for communication with the chassis
               DEVN, // device number
               period_us, // PWM period
               1, // number of channels to configure
               data_pwm); // arrayed structure of type pDQDIOPWM
                           // identifying PWM mode, channel to
                           // configure, and duty cycle or duration
    
```



**Figure 3-1 Disabling a DIO-432 Channel in Soft-both PWM Mode**

**3.4.2.1 Programming PWM Continuous Output**

When programming `DQDIO432_PWM_MODE` or `DQDIO432_PWM_MODE_GATED`, the `duty_cycle_length` parameter programs the PWM duty cycle instead of soft-start/stop modulation time duration.

In these PWM modes, `duty_cycle_length` resolution is 1/256. For the DIO-432, programming '0' results in 1/256% duty cycle and programming '255' results in a 100%. To set a 75% duty cycle, program '191' ( $256 \cdot 0.75 + 1$ ).



As an example, the following code configures an output line 10 to generate pulses continuously with a duty cycle equal to 75%:

```
// set the period of a single pulse period in microseconds
// programmed for all 32 dout channels (not per-channel programmable)

period_us = 10000; // PWM period

// set channel, mode and duty cycle (only 1 channel in data_pwm array)

data_pwm[0].mode = DQDIO432_PWM_MODE; // continuous
data_pwm[0].channel = 10; // set to DOut10 channel
data_pwm[0].duty_cycle_length = 191; // duty cycle= 191+1/256= 0.75

// set the period of the pulse width modulation in microseconds

DqAdv432SetPWM(hd, // handle for communication with the chassis
               DEVN, // device number
               period_us, // PWM period
               1, // number of channels to configure
               data_pwm); // arrayed structure of type pDQDIOPWM
                        // identifying PWM mode, channel to
                        // configure, and duty cycle or duration
```

**NOTE:** For the DIO-433, a '0' corresponds to 255/256% and 255 to 0%. Refer to the API description in the *PowerDNA API Reference Manual* and Sample432\_433 sample code for additional information.



**3.4.3 Writing Output Data** In Immediate mode, use the `DqAdv40xWrite()` API to write the ON/OFF state of all 32 DOut channels at once, as a port.

Provide `DqAdv40xWrite()` a bitwise mask of the ON/OFF state of each DOut channel:

- 0x00000000 set all 32 DOut channels OFF
- 0x00000001 set DOut0 ON, and DOut1..31 OFF
- 0x00000002 set DOut1 ON, and DOut0,DOut2..31 OFF
- 0x00000004 set DOut2 ON, and DOut0,DOut1,DOut3..31 OFF
- 0xFFFFFFFF set all 32 DOut channels ON

```
// switch ON channel 2 and 10
dataout = 0x000404;
DqAdv40xWrite(hd, DEVN, dataout);
```

**3.4.4 Reading Guardian Diagnostic Values** You have the option of reading the diagnostic channel current for each of the DOut channels on the DIO-432, DIO-432-800, DIO-433, and DIO-433-800.

Additionally, on the DIO-432 and DIO-433 (not -800 version), you have the option of performing a diagnostic read of the voltage.

Use the `DqAdv432GetAll()` API to read Guardian diagnostic values. The function returns 2 data structures: the data returned in the structure of type `pDQDIO432CVTD` provides converted floating point values for the current and voltage readings for each channel:

```
typedef struct {
    double current[32]; // Current value of the measured current (A)
    double voltage[32]; // Current value of the measured voltage (V)
} DQDIO432CVTD, *pDQDIO432CVTD;
```

To read the Guardian diagnostic data, do the following:

```
// declare arrays to hold returned data
// data_cvtd will hold the converted current and voltage data
DQDIO432DATAIN data_in;
DQDIO432CVTD data_cvtd;
```

```
// read Guardian diagnostic data
DqAdv432GetAll(hd, DEVN, &data_in, &data_cvtd);
```

Current reading for DOut0 will be in `data_cvtd.current[0]`; voltage reading for DOut0 will be in `data_cvtd.voltage[0]`, etc.



### 3.4.5 Reading Circuit

To monitor the status of the circuit breakers, you can use the `DqAdv432GetAll()` or `DqAdv40xRead()` API.

### Breaker Status & Reengaging

To reengage circuit breakers, use the `DqAdv432Reengage()` API (which was also described in Section 3.4.1).

To read the circuit breaker status with `DqAdv40xRead()`, do the following:

```
// declare arrays to hold returned data
uint32 cb_status[2];
```

```
// read the circuit breaker status
DqAdv40xRead(hd, DEVN, cb_status);
```

where

- `cb_status[0]` holds the current status of each breaker: 1 is tripped and 0 is not tripped
- `cb_status[1]` holds the sticky status of each breaker

To reengage tripped circuit breakers, use the `DqAdv432Reengage()` API and provide a bitmask of channels to reengage. If you want to reengage all the breakers that have tripped, you can provide `cb_status` bitmask:

```
// reset tripped circuit breakers
if(cb_status[0]) {
    DqAdv432Reengage(hd, DEVN, cb_status[0]);
}
```



# Appendix A

## A.1 Accessories

The following cables and STP boards are available for the DIO-432/433 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 x 1.2h 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.

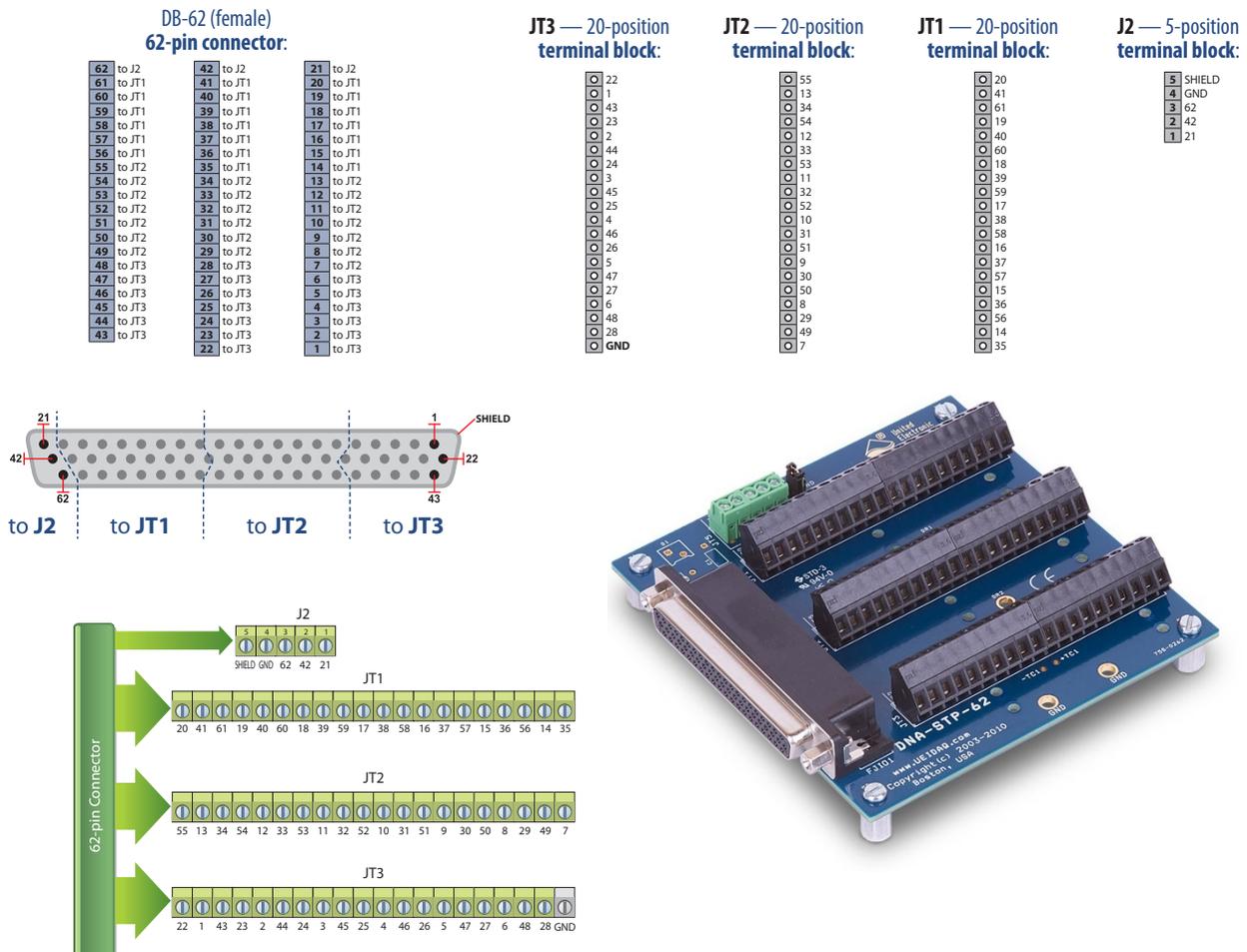


Figure A-1 Pinout and photo of DNA-STP-62 screw terminal panel



# Appendix B

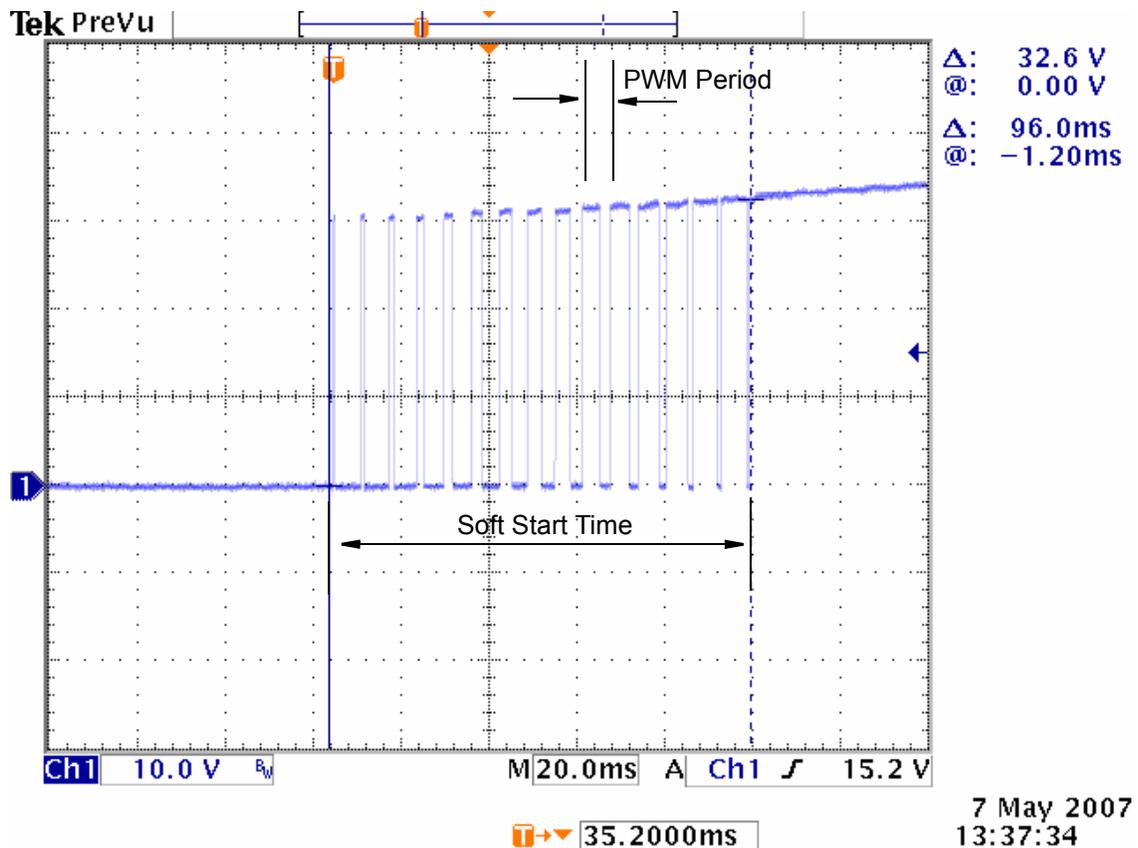
## Soft Start/Soft Stop/Constant PWM Feature Description

**B.1 PWM Soft Start** The DIO-432/433 series offer a per-channel configured pulse-width-modulated (PWM) start/stop mode. Refer to **Figure B-1** for a trace of a typical soft-start of the output (rising edge) configured for a transition of 16 PWM period steps with the period set to approximately 6 mS.

This feature is useful in preventing premature burnout of devices such as incandescent bulbs caused by too rapid heating on startup.

Note that the on/off duty cycle (ratio of on time to off time during a PWM period) increases progressively from minimum to maximum during the start-up time. The rate at which the duty cycle increases is determined by a software parameter.

For programming examples using UEI Framework API, refer to **Chapter 2**, Section 2.6 on page 20, or refer to **Chapter 3**, Section 3.4.2 on page 27 for low-level C programming information.



**Figure B-1. Typical Soft Start Cycle on DIO-433**

7 May 2007  
13:37:34



**B.2 PWM Soft Stop** The DIO-432/433 series can also be configured for soft stop mode of operation, which is the reverse of a soft start operation. The PWM duty cycle decreases gradually from maximum to minimum over the Soft Stop Time. The typical application for soft stop mode is a soft start operation that is implemented with inverted logic.

**B.3 PWM Continuous Mode** The constant PWM mode of operation has a constant duty cycle in which the ratio of on time to off time is fixed for the entire period of operation. The ratio value may be set as a software parameter. A typical application for this feature is a dimmer for an incandescent indicator light in which the average voltage applied to a bulb is increased or decreased by varying the PWM duty cycle.

**B.4 Programming Notes** The code below is an excerpt from `Sample432_433.c` found in the low-level examples folder (refer to Section 3.3 on page 25 for sample code location):

```
// The following is an example of code for implementing the
// soft start feature for: D=W/T (16PWM)

#define IOM_IPADDR0          "192.168.100.2"
#define DEVN                 0
#define CHANNELS             DQ_DIO432_CHAN
#define PWM_MODE             1 // 1-enable/0-disable PWM mode (Optional mode)
#define PWM_MODE_DUTY_CYCLE (8).
```

When in continuous `PWM_MODE`, users can control the duty cycle of the modulated output. For details, see NOTES 2 and 3 below.

In soft start/soft stop modes, (`DQDIO432_PWM_SOFTSTART`, `DQDIO432_PWM_SOFTSTOP`, and `DQDIO432_PWM_SOFTBOTH`), the duty cycle variable is used to set the number of PWM periods. For example, if `<duty_cycle == 16>` the output produces 16 PWM periods upon changing state, from 1/16 duty cycle to 16/16 (or 100%). Refer to **Figure B-1**.

**NOTE:**

1. To avoid confusion when setting up a PWM operating mode, note that the DNx-DIO-432 output circuit has a FET to ground on each output. Therefore, an ON state is a signal LOW. Conversely, the DNx-DIO-433 has a FET to +Vcc on each output. Therefore, an ON state for the DIO-433 is a signal HIGH.
2. For the DIO-432:  
To attain 0% duty cycle, set the `PWM_MODE` for the channel to `DQDIO432_PWM_DISABLED` and set the corresponding output to 0 using the `DqAdv40xWrite()` function (if programming using the low-level C-based API).
3. For the DIO-433: a `DUTY_CYCLE_LENGTH` of 0 corresponds to 255/256% (99.6%) and 255 to 0%. You cannot directly set the duty cycle to 100% on a DIO-433.  
To attain 100% duty cycle with a DIO-433, set the `PWM_MODE` for the channel to `DQDIO432_PWM_DISABLED` and set the corresponding output to 1 using the `DqAdv40xWrite()` function.



# Index

## A

Acquisition modes 26  
ADC Conversion Speed 11  
Analog Voltage Input 4

## B

Block diagram 10

## C

Cable(s) 32  
Circuit Breakers 23, 31  
Cleaning-up the Session 21, 22  
Cleaning-up the session 21, 22  
Configuring the Circuit Breaker 11  
Configuring the Resource String 18  
Conventions 2  
Creating a Session 18  
Current Monitoring 10  
Current Sense Resistors 3

## H

High Level API 18

## L

Low-level API 24

## O

Organization 1

Overcurrent Response Time 6

## P

Physical Description 10  
Programming Circuit Breakers 23  
PWM Mode 34

## S

Schematic for High-Side FET 4  
Schematic for Low-Side FET 4  
Screw Terminal Panels 32  
Setting Operating Parameters 7  
Soft Stop 34  
Support ii  
Support email  
    [support@ueidaq.com](mailto:support@ueidaq.com) ii  
Support FTP Site  
    <ftp://ftp.ueidaq.com> ii  
Support Web Site  
    [www.ueidaq.com](http://www.ueidaq.com) ii

## T

Technical Specifications for the DIO-433 8

## W

Wiring 14