

# **DNx-DMM-261**

## **User Manual**

**6 $\frac{1}{2}$ -digit Digital Multimeter Board  
for the PowerDNA Cube and RACK series chassis**

**October 2023**

**PN Man-DNx-DMM-261**

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, by photocopying, recording, or otherwise without prior written permission.

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:

249 Vanderbilt Avenue  
Norwood, MA 02062  
U.S.A.

### Shipping Address:

24 Morgan Drive  
Norwood, MA 02062  
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 Manual Conventions .....	2
1.3 Naming Conventions .....	2
1.3.1 UEI Product Names .....	2
1.3.2 Channel Naming .....	2
1.4 Related Resources .....	3
1.5 Before You Begin .....	4
1.6 DNx-DMM-261 Features .....	5
1.6.1 Measurement Capabilities .....	5
1.6.2 Noise Filtering .....	5
1.6.3 Guardian Diagnostics .....	6
1.6.4 Input Protection .....	6
1.6.5 Isolation .....	6
1.6.6 Environmental Conditions .....	6
1.6.7 Accessories .....	6
1.6.8 Software Support .....	7
1.7 Technical Specifications .....	8
<b>Chapter 2 Functional Description</b> .....	<b>10</b>
2.1 Device Architecture .....	10
2.2 DC Voltage (DCV) Measurement .....	11
2.2.1 DCV Range Selection .....	12
2.2.2 DCV Input Protection .....	12
2.2.3 DCV Auto-zero .....	13
2.2.4 DCV Averaging .....	13
2.3 AC Voltage (ACV) Measurement .....	13
2.3.1 ACV Range Selection .....	14
2.3.2 ACV Input Protection .....	14
2.3.3 ACV Zero Crossing Detection .....	15
2.3.4 ACV True RMS Conversion .....	15
2.3.5 ACV Frequencies .....	15
2.3.6 ACV Peak Values .....	16
2.4 DC Current (DCI) Measurement .....	16
2.4.1 DCI Range Selection .....	17
2.4.2 DCI Input Protection .....	17
2.4.3 DCI Auto-zero .....	17
2.4.4 DCI Averaging .....	18
2.5 AC Current (ACI) Measurement .....	18
2.5.1 ACI Range Selection .....	19
2.5.2 ACI Input Protection .....	19
2.5.3 ACI Zero Crossing Detection .....	20
2.5.4 ACI True RMS Conversion .....	20
2.5.5 ACI Frequencies .....	20
2.5.6 ACI Peak Values .....	20



2.6	Power Measurement . . . . .	21
2.7	Resistance Measurement . . . . .	21
2.7.1	2-Wire Resistance (R2W) . . . . .	21
2.7.2	4-Wire Resistance (R4W) . . . . .	22
2.7.3	Resistance Ranges . . . . .	24
2.7.4	Resistance Input Protection . . . . .	24
2.8	Guardian Diagnostics . . . . .	25
2.8.1	Voltage Monitor . . . . .	25
2.8.2	Current Monitor . . . . .	25
2.9	Indicators and Connectors . . . . .	25
2.10	Pinout . . . . .	26
2.11	Multiplexer Boards for the DMM-261 . . . . .	28
2.11.1	Connecting DMM and MUX Boards within a Cube . . . . .	28
2.11.2	Connecting DMM and MUX Boards within a Rack . . . . .	28
<b>Chapter 3 PowerDNA Explorer . . . . .</b>		<b>31</b>
3.1	Introduction . . . . .	31
3.2	Read DMM Data . . . . .	32
3.2.1	DC Voltage and Current . . . . .	33
3.2.2	AC Voltage and Current . . . . .	33
3.2.3	Overrange Error . . . . .	33
3.2.4	Protection Tripped Indicator . . . . .	34
3.2.5	Resistance Channel . . . . .	34
3.2.6	Auto-ranging . . . . .	34
3.3	Configuration with MUX-461 Boards . . . . .	35
<b>Chapter 4 Programming with the High-level API . . . . .</b>		<b>36</b>
4.1	About the High-level API . . . . .	36
4.2	Example Code . . . . .	36
4.3	DMMChannel Methods . . . . .	37
4.4	Create a Session . . . . .	37
4.5	Resource Strings . . . . .	37
4.6	Configure DMM Channels . . . . .	38
4.6.1	DC Measurement . . . . .	38
4.6.2	AC Measurement . . . . .	40
4.6.3	Resistance Measurement . . . . .	41
4.6.4	Auto-Ranging . . . . .	42
4.6.5	Guardian Diagnostic Measurement . . . . .	42
4.6.6	Add Time Stamp . . . . .	42
4.6.7	Configure Input Disconnection . . . . .	43
4.7	Configure the Timing . . . . .	43
4.8	Start the Session . . . . .	43
4.9	Read Data . . . . .	43
4.9.1	Read Input Protection Status . . . . .	44
4.10	Stop the Session . . . . .	45



4.11 Using the DMM-261 with the MUX-461 . . . . . 45

**Chapter 5 Programming with the Low-level API . . . . . 47**

5.1 About the Low-level API . . . . . 47

5.2 Example Code . . . . . 47

5.3 Data Acquisition Modes . . . . . 48

5.4 Point-by-Point API . . . . . 48

**Appendix A Accessories . . . . . 50**

A.1 Cables . . . . . 50

A.2 Screw Terminal Panels . . . . . 50



# List of Figures

<b>Chapter 1 Introduction</b> .....	<b>1</b>
<b>Chapter 2 Functional Description</b> .....	<b>10</b>
2-1 High-level Block Diagram of the DNx-DMM-261 Board .....	11
2-2 Simplified Block Diagram for DC Voltage .....	11
2-3 Connections for DC Voltage Measurement .....	12
2-4 Simplified Block Diagram for AC Voltage .....	13
2-5 Connections for AC Voltage Measurement .....	14
2-6 Simplified Block Diagram for DC Current .....	16
2-7 Connections for DC Current Measurement .....	16
2-8 Simplified Block Diagram for AC Current.....	18
2-9 Connections for AC Current Measurement.....	19
2-10 Connections for Power Measurement .....	21
2-11 Connections for 2-Wire Resistance Measurement .....	22
2-12 Simplified Block Diagram for 2-Wire Resistance .....	22
2-13 Connections for 4-Wire Resistance Measurement .....	23
2-14 Simplified Block Diagram for 4-Wire Resistance .....	23
2-15 Photo of DNR-DMM-261 Board.....	25
2-16 Pinout Diagram for the DNx-DMM-261 .....	26
2-17 DNR-DMM-261 (left) connected to five DNR-MUX-461 Boards .....	29
2-18 Installing DMM-MUX Cable .....	30
<b>Chapter 3 PowerDNA Explorer</b> .....	<b>31</b>
3-1 PowerDNA Explorer for DNx-DMM-261 .....	32
3-2 Example of Overrange Error.....	33
3-3 Protection Tripped Indicator .....	34
3-4 Auto-ranging Indicator .....	34
3-5 Configuration with DMM-261 and MUX-461 Boards.....	35
<b>Chapter 4 Programming with the High-level API</b> .....	<b>36</b>
<b>Chapter 5 Programming with the Low-level API</b> .....	<b>47</b>
<b>Appendix A Accessories</b> .....	<b>50</b>
A-1 Pinout and Photo of DNA-STP-37 Screw Terminal Panel.....	51



# List of Tables

<b>Chapter 1 Introduction</b>	<b>1</b>
1-1 Specifications Overview	8
1-2 Detailed Input Specifications	9
<b>Chapter 2 Functional Description</b>	<b>10</b>
2-1 DCV Ranges ( $\pm$ )	12
2-2 ACV Ranges ( $\pm$ )	14
2-3 DCI Ranges ( $\pm$ )	17
2-4 ACI Ranges ( $\pm$ )	19
2-5 Resistance Ranges	24
2-6 DNx-DMM-261 LED Indicators	26
2-7 Signal Connections for the DNx-DMM-261	27
<b>Chapter 3 PowerDNA Explorer</b>	<b>31</b>
<b>Chapter 4 Programming with the High-level API</b>	<b>36</b>
4-1 CUiDMMChannel Class Methods	37
4-2 CUiDMMReader Class Methods	43
<b>Chapter 5 Programming with the Low-level API</b>	<b>47</b>
5-1 Low-level API for the DNx-DMM-261	49
5-2 MuxGroup API Functions	49
<b>Appendix A Accessories</b>	<b>50</b>



# Chapter 1 Introduction

This manual outlines the feature set and use of the DNx-DMM-261, a 6½-digit Digital Multimeter (DMM) Board that brings the performance and functionality of a stand-alone DMM to UEI's popular rugged, easily distributed chassis series.

The following sections are provided in this chapter:

- Organization of this Manual (Section 1.1)
- Manual Conventions (Section 1.2)
- Naming Conventions (Section 1.3)
- Related Resources (Section 1.4)
- Before You Begin (Section 1.5)
- DNx-DMM-261 Features (Section 1.6)
- Technical Specifications (Section 1.7)

## 1.1 Organization of this Manual

This DNx-DMM-261 User Manual is organized as follows:

- **Introduction**  
Chapter 1 summarizes the features and specifications of the DNx-DMM-261.
- **Functional Description**  
Chapter 2 describes the device architecture, logic, and connectivity of the DNx-DMM-261.
- **PowerDNA Explorer**  
Chapter 3 shows how to explore DNx-DMM-261 features through a GUI-based application.
- **Programming with the High-level API**  
Chapter 4 describes how to create a session, configure the session, and interpret results with the Framework API.
- **Programming with the Low-level API**  
Chapter 5 provides an overview of programming the DNx-DMM-261 using low-level C API.
- **Accessories**  
Appendix A provides a list of accessories available for use with the DNx-DMM-261.



## 1.2 Manual Conventions

The following conventions are used throughout this manual:



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



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

**NOTE:** Notes alert you to important information.

Typeface	Description	Example
<b>bold</b>	field or button names	Click <b>Scan Network</b>
<b>»</b>	hierarchy to get to a specific menu item	<b>File » New</b>
fixed	source code to be entered verbatim	<code>session.CleanUp()</code>
<brackets>	placeholder for user-defined text	<code>pdna://&lt;IP address&gt;</code>
<i>italics</i>	path to a file or directory	<i>C:\Program Files</i>

## 1.3 Naming Conventions

This user manual will use the naming conventions described below.

### 1.3.1 UEI Product Names

The DNA-DMM-261, DNR-DMM-261, and DNF-DMM-261 board versions are compatible with the UEI Cube, RACKtangle, and FLATRACK chassis respectively. These boards are electronically identical and differ only in mounting hardware. The DNA version stacks in a Cube chassis, while the DNR and DNF versions plug into the backplane of a Rack chassis. Throughout this manual, the term DNx-DMM-261 refers to both Cube and Rack products.

Additionally, the DNx-MUX-461 and DNx-MUX-461-350 multiplexer boards that are available as add-on options (see Section 1.6.7.1) for the DNx-DMM-261 will be referred to using “DNx-MUX-461” or “DNx-MUX-461 Series” throughout this manual except to distinguish differences in functionality.

### 1.3.2 Channel Naming

Throughout this user manual, the term “channel” will be used in the following contexts:

- *measurement channels* - represent internal DMM-261 registers where data is read from. This may appear in the form of the channel name used in the low-level API, e.g., “DQ\_DMM261\_RES\_CH”, or a textual description, e.g., “resistance channel”.
- *DMM channel* - refers to the UEI Framework class (`CUEiDMMChannel`) that manages settings for each DMM input channel. UEI Framework channel classes provide access to the subsystems used by the Framework to communicate with UEI hardware.



In the UEI Framework, a measurement mode is specified when creating a DMM channel. The measurement mode in the Framework corresponds to a measurement channel in the low-level API.

- *MUX Input channel (or Input channel)* - refers to an input from a DNx-MUX-461 Series multiplexer board.

## 1.4 Related Resources

This manual only covers functionality specific to the DNx-DMM-261. To get started with the UEI IOM, please see the documentation included with the software installation. On Windows, these resources can be found from the desktop by clicking **Start » All Programs » UEI**

UEI's website includes other user resources such as application notes, FAQs, tutorials, and videos. In particular, the glossary of terms may be helpful when reading through this manual: <https://www.ueidaq.com/glossary>

Additional questions? Please email UEI Support at [support@ueidaq.com](mailto:support@ueidaq.com) or call 508-921-4600.



## 1.5 Before You Begin



### ***No Hot Swapping!***

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.



### ***Check Your Firmware***

Ensure that the firmware installed on the Cube or Rack CPU matches the UEI software version installed on your PC. The IOM is shipped with pre-installed firmware and a matching software installation. If you upgrade your software installation, you must also update the firmware on your Cube or RACK CPU. See "*Firmware Update Procedures.pdf*" for instructions on checking and updating the firmware. This document is located in the following locations:

- On Linux: *PowerDNA\_Linux\_<x.y.z>/docs*
- On Windows:  
**Start » All Programs » UEI » DNx Firmware Update Procedures**



**1.6 DNx-DMM-261 Features** The DNx-DMM-261 Digital Multimeter Board brings full 6.5 digit performance to UEI's rugged, easily distributed chassis. When used with DNx-MUX-461 multiplexer boards, up to 130 MUX input channels can be acquired in a single chassis by a single DMM (see Section 1.6.7.1). Features of the DNx-DMM-261 include:

- 6.5 digit resolution
- Measures VDC, VAC, IDC, IAC, and Resistance
- True RMS measurement
- -40 °C to +85 °C operating temperature range
- 350 Vrms isolation

**1.6.1 Measurement Capabilities** The DNx-DMM-261 can measure the following types of inputs:

- DC Voltage
  - Max input range  $\pm 300$  V DC
  - Min input range  $\pm 0.1$  V DC
- AC Voltage (20 Hz - 3 kHz)
  - Max input range  $\pm 300$  Vrms (true RMS)
  - Min input range  $\pm 0.50$  Vrms (true RMS)
- DC Current
  - Max input range  $\pm 2$  A DC
  - Min input range  $\pm 1.5$  mA DC
- AC Current (20 Hz - 3 kHz)
  - Max input range  $\pm 2.0$  Arms (true RMS)
  - Min input range  $\pm 1.5$  mArms (true RMS)
- DC and AC Power
- Resistance Measurement
  - 2-wire or 4-wire configuration
  - Max input range: 100 M $\Omega$  (2-wire measurements)
  - Max input range: 10 M $\Omega$  (4-wire measurements)
  - Min input range: 10  $\Omega$

Measurement channels may be configured to auto-range or use a manually selected range. Note that when using the UeiDaq Framework API, auto-ranging is enabled by default.

**1.6.2 Noise Filtering** The DNx-DMM-261 is equipped with noise reduction features designed to increase measurement accuracy in noisy environments.

- Anti-aliasing low-pass RC filter on the front-end
- Configurable sample averaging engine
- AC power line noise rejection (NPLC)



- 1.6.3 Guardian Diagnostics** In addition to the standard V, I, and R measurement channels, the DNx-DMM-261 provides diagnostic channels for reading back on-board temperature, internal supply voltages, and current. Diagnostic data is retrieved from independent health monitoring ADCs.
- 1.6.4 Input Protection** Voltage and resistance inputs are protected with a configurable automatic disconnect feature. When enabled, inputs are automatically disconnected in the event of a DC over-voltage. If a higher range is available, the firmware automatically increases the range and reconnects the inputs.
- Current inputs are protected with a 2.5 A fuse and the current monitor ADC. When over-current conditions are detected, the board automatically switches to the highest range.
- For DCV and ACV inputs, if the board was already in the maximum range at the time of an over-voltage condition, the safety relay remains open until the user reconfigures or power cycles the board.
- 1.6.5 Isolation** The DNx-DMM-261 offers 350 Vrms of isolation between itself and other I/O boards as well as between the I/O connections and the chassis.
- 1.6.6 Environmental Conditions** Like all UEI I/O boards, the board offers operation in extreme environments and has been designed to operate to 5 g vibration and 100 g shock. The board has been tested from -40 to +85 °C and will function at altitudes up to 70,000 feet.
- 1.6.7 Accessories** UEI offers the following accessories for use with DNx-DMM-261 digital multimeter boards.
- 1.6.7.1 Multiplexers** The following multiplexer boards are available for purchase as add-on options with the DNx-DMM-261:
- The DNx-MUX-461 Multiplexer Board is designed to provide up to 26 two-wire or up to 13 four-wire channels to the DNx-DMM-261. Input channels to the DMM-261 from the DNx-MUX-461 may be a mixture of two and four-wire measurements.
  - The DNx-MUX-461-350 is a high voltage multiplexer board that provides up to 24 two-wire or up to 12 four-wire channels to the DNx-DMM-261 and is capable of switching voltages up to 350 V DC or AC. Input channels to the DMM-261 from the DNx-MUX-461-350 may be a mixture of two and four-wire measurements.

All DMM connections are made inside the Cube or RACKtangle so the only connections you need to make are to the multiplexer input channels. Up to five multiplexer boards may be daisy-chained together to provide up to 5x the number of MUX input channels for your configuration. Note that both DNx-MUX-461 and DNx-MUX-461-350 multiplexers may be included in the same configuration.

In a Cube chassis, a set of internal connectors connect the DNx-DMM-261 and MUX-461 Series boards. In a RACKtangle chassis or HalfRACK chassis, a DNR-CBL-461-x cable is used. See Section 2.11.2 for information on connecting DMM-261 boards to DNR-MUX-461 Series boards in a Rack chassis.



### 1.6.7.2 Cables and STPs

All field-wiring connections to the DNx-DMM-261 are made through a standard 37-pin D connector. The following cabling options are available from UEI:

- **DNA-CBL-261-BNC** cables are 12 inches long and connect to field devices using standard BNC connectors. DNA-CBL-261-BNC cables are recommended for the best performance.
- **DNA-CBL-261** cables are 10 inches long and connect to field devices using standard banana-style connectors.
- **DNA-CBL-37s** cables are 37-way, male to female, round shielded cables that can be ordered in lengths of 1, 3, 5, 10, or 20 feet.

Users may also connect the DNx-DMM-261 board to UEI's DNA-STP-37 screw terminal panel via **DNA-CBL-37** flat ribbon cables.

Additionally, OEM users can build custom cabling systems through off-the-shelf components.

### 1.6.8 Software Support

The DNx-DMM-261 includes a software suite supporting Windows, Linux, QNX, VxWorks, RTX, and most other popular real-time operating systems. Windows users may use the UeiDaq Framework, which provides a simple and complete software interface to Windows programming languages and DAQ applications (e.g., LabVIEW, MATLAB). All software support includes extensive example programs that make it easy to copy-and-paste the I/O software into your applications.



## 1.7 Technical Specifications

**Table 1-1** summarizes the technical specifications for the DNx-DMM-261 board. Detailed input specifications are provided in **Table 1-2**. All specifications are at 23°C±5°C unless otherwise noted.

**Table 1-1 Specifications Overview**

Input Measurement Ranges	
DC Voltage	
Max input range	± 300 V DC
Min input range	± 0.1 V DC
AC Voltage (20 Hz - 3 kHz)	
Max input range	± 300 Vrms (true RMS)
Min input range	± 0.5 Vrms (true RMS)
DC Current	
Max input range	± 2 A DC
Min input range	± 1.5 mA DC
AC Current (20 Hz - 3 kHz)	
Max input range	± 2 Arms (true RMS)
Min input range	± 1.5 mArms (true RMS)
Resistance Measurement	
Configuration	2-wire or 4-wire
Max input range	100 MW for 2-wire measurements 10 MΩ for 4-wire measurements
Min input range	10 Ω
General Specifications	
Isolation	350 Vrms
Operating temperature	Tested -40°C to +85°C
Storage temperature	-55°C to +100°C
Vibration IEC 60068-2-6 IEC 60068-2-64	5 g, 10-500 Hz, sinusoidal 5 g (rms), 10-500 Hz, broadband random
Shock IEC 60068-2-27	100 g, 3 ms half sine, 18 shocks @ 6 orientations 30 g, 11 ms half sine, 18 shocks @ 6 orientations
Operating humidity	0% to 95%, non-condensing
Power consumption	3.5 W max
MTBF	450,000 hours



**Table 1-2 Detailed Input Specifications**

**Note:** all specs at 23 °C ±5 °C unless otherwise noted

**DC Volt ranges**

+/- Volts	Rin, MΩ	Calibration Accuracy ppm	Calibration Accuracy mV	Gain drift ppm/°C	Offset drift ppm/°C	Linearity ppm
300	10	20	6.0	8	5	6
100	10	30	3.0	8	5	6
10	1000+	15	0.15	4	2	4
1	1000+	15	0.015	4	2	4
0.1	1000+	15	0.015			

**AC Volt ranges (20Hz - 3 kHz)**

Vrms	Rin, MΩ	Calibration Accuracy ppm@60Hz	Calibration Accuracy mVrms
300		1250	375
100		2500	250
10		7500	75
1.5		25,000	38
0.5		20,000	10

Note: Vrms calculated based on the data acquired at 1.6MS/sec averaged over TBD periods

**DC Current ranges (+ at least 20% overrange)**

+/-I, A	Shunt, Ω	Vshunt	Calibration Accuracy %	Calibration Accuracy, mA	Gain drift ppm/°C	Offset drift ppm/°C	Linearity ppm
2	0.1	0.3	0.17	5.0	50	2	4
1.5	0.1	0.15	0.33	5.0	50	2	4
0.15	0.1	0.015	0.30	0.45	20	2	4
0.015	10	0.15	0.30	0.05	20	2	4
0.0015	10	0.015	0.20	0.02	10	2	4

**AC Current ranges (20Hz - 3 kHz) + at least 20% overrange for all but 2A**

Arms	Shunt, Ω	Vshunt	Calibration Accuracy %	Calibration Accuracy, mArms	Gain drift ppm/°C	Offset drift ppm/°C	Linearity ppm
2	0.1	0.2	0.075	5.0	50	2	4
1	0.1	0.15	0.15	3.0	50	2	4
0.15	0.1	0.015	0.33	1.0	20	2	4
0.015	10	0.15	3.3	0.5	20	2	4
0.0015	10	0.015	0.067	0.1	10	2	4

**Resistance 2- and 4- wire (19.7Hz with 50/60Hz rejection)**

Range, Ω	2-Wire Accuracy Ω	4-Wire Accuracy Ω	Gain drift ppm/°C	Offset drift ppm/°C	Linearity ppm
10	1	0.012	100	5	10
100	1	0.015	25	5	10
1k	1	0.05	15	2	10
10k	1.5	0.50	15	2	10
100k	6	6.0	15	2	10
1M	101	101	15	2	10
10M	25k	25k	15	2	10
100M	1 Meg	n/a	15	2	10



## Chapter 2 Functional Description

This chapter describes the device architecture, hardware, and functionality of the DNx-DMM-261 Digital Multimeter Board. The following sections are provided in this chapter:

- Device Architecture (Section 2.1)
- DC Voltage (DCV) Measurement (Section 2.2)
- AC Voltage (ACV) Measurement (Section 2.3)
- DC Current (DCI) Measurement (Section 2.4)
- AC Current (ACI) Measurement (Section 2.5)
- Power Measurement (Section 2.6)
- Resistance Measurement (Section 2.7)
- Guardian Diagnostics (Section 2.8)
- Indicators and Connectors (Section 2.9)
- Pinout (Section 2.10)
- Multiplexer Boards for the DMM-261 (Section 2.11)

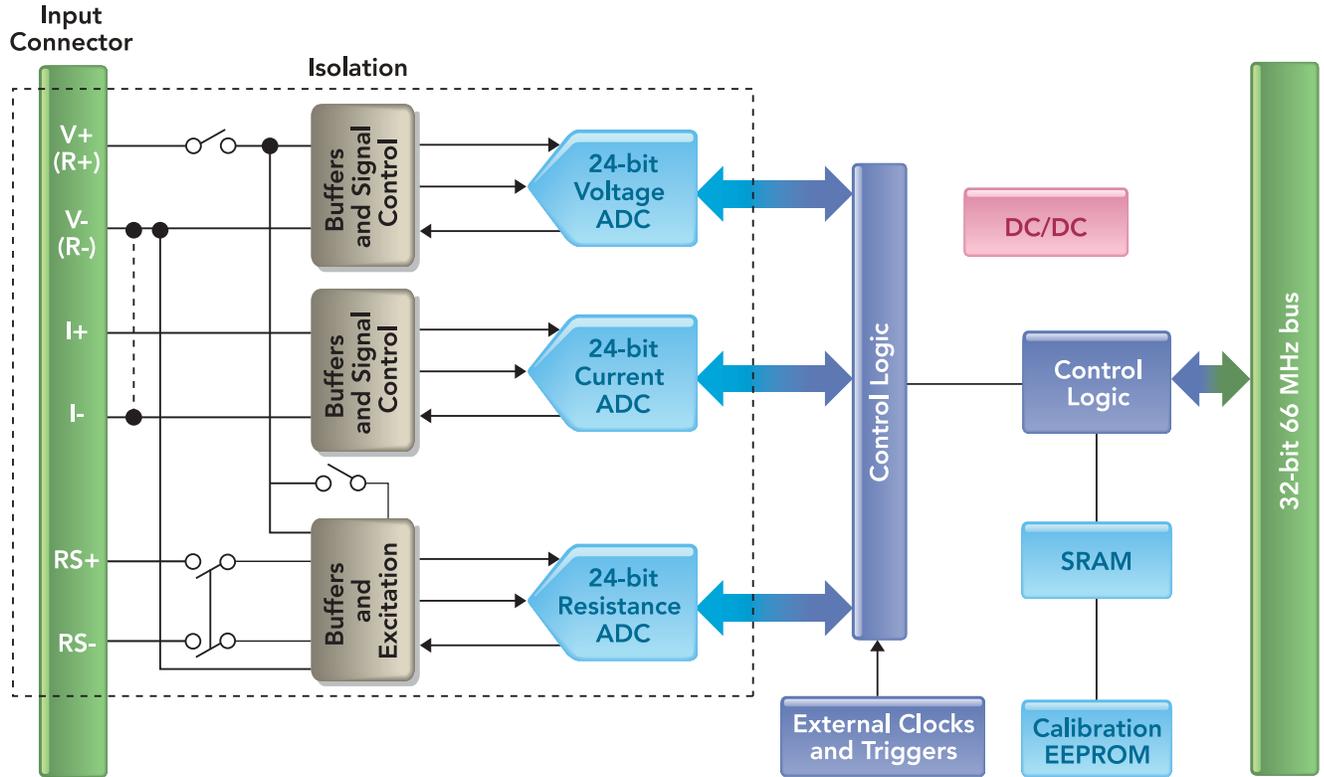
### 2.1 Device Architecture

The DNx-DMM-261 utilizes five ADCs, a DAC, relays, and other digital circuitry in order to cover all required ranges and modes of operations. As shown in **Figure 2-1**, external inputs first pass through front-end protection circuitry. This allows inputs to be disconnected in the event of over-voltage or over-current. Input signals then undergo amplification and hardware noise filtering before A/D conversion.

Three of the five ADCs are dedicated to Voltage, Current, and Resistance measurements respectively, and the remaining two ADCs are used to monitor board health. Reference signals for A/D conversion are derived from an ultra-precise buffered reference (5.0 V, 1 ppm). The control logic can optionally filter and average digitized signals before returning the data to the user application. Note that all I/O circuitry is optically isolated from the control logic. Additionally, the DNx-DMM-261 board is fully isolated from the Cube or RACK chassis.



Please refer to Sections 2.2 through 2.7 for more details on the design of the Voltage, Current, and Resistance subsystems.

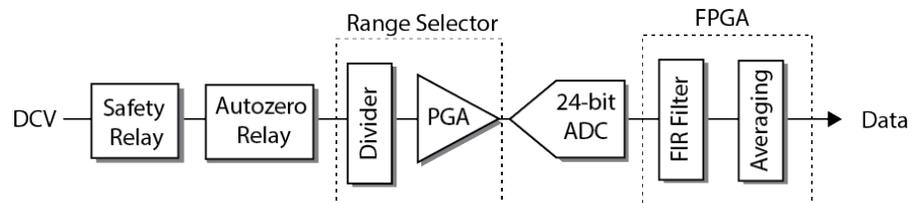


**Figure 2-1 High-level Block Diagram of the DNx-DMM-261 Board**

## 2.2 DC Voltage (DCV) Measurement

The DNx-DMM-261 can measure DC voltages between 30 mV and 300 V.

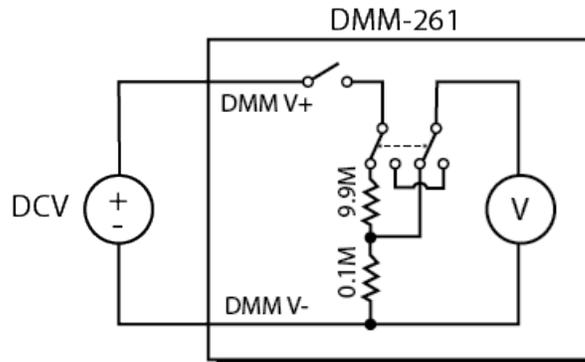
**Figure 2-2** illustrates the main functional blocks in the DCV measurement path. The design centers around a high-speed (1.6 MS/sec) 24-bit simultaneous sampling auto-averaging ADC with a PGA front end. Note that some DCV hardware is shared with the ACV subsystem.



**Figure 2-2 Simplified Block Diagram for DC Voltage**

The circuit for a typical DCV measurement is shown in **Figure 2-3**. The front-end switch is used for over-voltage protection (Section 2.2.2) and auto-zeroing (Section 2.2.3). The 10 MΩ divider is activated in the high voltage ranges (see Section 2.2.1).





**Figure 2-3 Connections for DC Voltage Measurement**

**2.2.1 DCV Range Selection**

The DNx-DMM-261 supports the DCV ranges listed in **Table 2-1**.

**Table 2-1 DCV Ranges (±)**

Range	Under-range	Overrange	Over-voltage
300 V	50 V	n/a	350 V
100 V	5 V	120 V	124 V
10 V	0.5 V	12 V	13 V
1 V	0.05 V	1.2 V	2.9 V
0.1 V	0.015 V	0.12 V	0.36 V

Users can either select a range manually or configure the DCV measurement channel for auto-ranging. Auto-ranging is useful for measuring a signal with an unknown or varying range. In auto-ranging mode, the DMM automatically switches the range up if the input is overrange or down if the input is under-range (**Table 2-1**). The board maximizes resolution by returning data from the smallest acceptable range. It also returns the range used for the measurement.

When a DCV range is programmed, the firmware selects a gain on the PGA and enables or disables a 100:1 voltage divider. The divider is shown in **Figure 2-3**. In the 10 V and under ranges, the divider is disabled and the input resistance is >1 GΩ. In the 100 V and 300 V ranges, the divider is used to bring the voltage levels within the upper limits of the ADC. Enabling the divider reduces the input resistance to 10 MΩ.

**2.2.2 DCV Input Protection**

A high-speed safety relay provides automatic disconnect protection in case of DC over-voltage. Input voltages are continuously compared to over-voltage limits (range-specific limits are listed in **Table 2-1**). When protection trips, the firmware automatically disconnects the input and reconnects in the maximum range. If the board was already in the maximum range, the safety relay remains open until the user reconfigures or power cycles the board.



In manual ranging mode, “auto-range” and “overrange” status flags are set to let the user know about the range change. Once the data falls within the limits of the user configured range, the board will re-enter the user’s range.

The status of the input protection system can be read by the user application.

**2.2.3 DCV  
 Auto-zero**

Users can auto-zero DC voltage measurement channels to correct for any offset voltages in the measurement path. When the user calls the low-level API function `DqAdv261AutoZero()`, the auto-zero relay disconnects the external inputs from the DMM input circuitry. The board takes an average zero value reading over the specified duration and then reconnects the inputs. The new zero values are automatically subtracted from future measurements. Auto-zero can be enabled in the Framework API by calling `EnableAutoZero()`.

**2.2.4 DCV  
 Averaging**

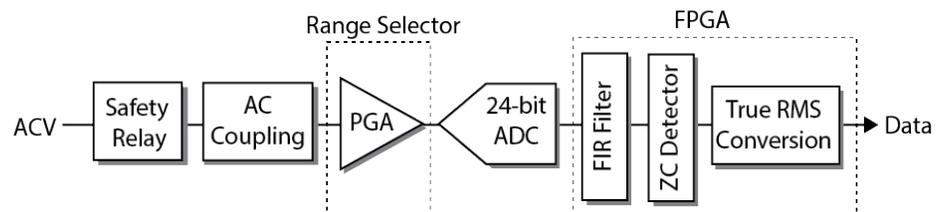
Samples from the ADC can be buffered and averaged to achieve higher resolution and accuracy. After the specified number of samples have accumulated in the memory stack, their average value is returned to the user and the stack is cleared. The following averaging modes are supported for DCV measurements:

- Averaging OFF: Returns the calibrated DC voltage with no sample averaging.
- Averaging ON: This accumulate and divide mode averages all DCV samples acquired since the DCV measurement channel was last read. This mode is ideal for applications where data is read infrequently.
- NPLC Averaging: Averages DCV samples over a configurable number of power line cycles. This mode is useful for eliminating AC power line noise, since an actual power line sine wave averages out to near zero over a whole number of periods.

**2.3 AC Voltage  
 (ACV)  
 Measurement**

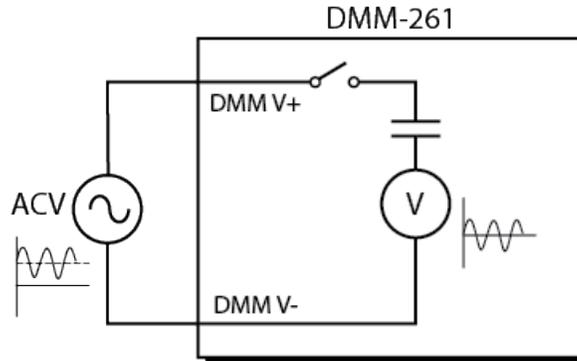
The DNx-DMM-261 can measure true RMS AC voltages within several user-selectable ranges. The smallest range is  $\pm 0.5$  Vrms and the largest range is  $\pm 300$  Vrms. See **Table 2-2** for more detail and a list of available ranges.

**Figure 2-4** illustrates the main functional blocks in the ACV measurement path. The ACV subsystem shares several hardware components with the DCV subsystem including the safety relay, high-speed (1.6 MS/sec) 24-bit simultaneous sampling auto-averaging ADC, and PGA.



**Figure 2-4 Simplified Block Diagram for AC Voltage**

A typical ACV measurement circuit and sample input waveform is shown in **Figure 2-5**. The safety relay protects against over-voltage (see Section 2.3.2), and a coupling capacitor blocks the DC component of the input signal. This allows the DMM to measure AC signals which occur on top of large DC offsets, such as ripples on a power supply. All RMS and peak values are measured from the original DC value, not from DC zero.



**Figure 2-5 Connections for AC Voltage Measurement**

**2.3.1 ACV Range Selection**

The DNx-DMM-261 supports the AC voltage ranges listed in **Table 2-2**.

**Table 2-2 ACV Ranges (±)**

Range	Under-range	Overrange	Over-voltage
300 Vrms	50 Vrms	n/a	400 V
100 Vrms	5 Vrms	110 Vrms	150 V
10 Vrms	1 Vrms	12 Vrms	19 V
1.5 Vrms	0.4 Vrms	1.55 Vrms	2.2 V
0.5 Vrms	n/a	0.77 Vrms	1.1 V

Users can either select a range manually or configure the ACV measurement channel for auto-ranging. Auto-ranging is useful for measuring a signal with an unknown or varying range. In auto-ranging mode, the DMM automatically switches the range up if the input is overrange or down if the input is under-range (**Table 2-2**). The board maximizes the resolution by returning data from the smallest acceptable range. It also returns the range used for the measurement.

**2.3.2 ACV Input Protection**

A high-speed safety relay disconnects ACV inputs if a single ADC reading is above the over-voltage limit for the configured range. These limits are listed in **Table 2-2**. Note that this is an actual ADC reading, not an RMS or peak value. When protection trips, the firmware automatically disconnects the input and reconnects in the maximum range. If the board was already in the maximum range, the safety relay remains open until the user reconfigures or power cycles the board.



In manual ranging mode, “auto-range” and “overrange” status flags are set to let the user know about the range change. Once the peak voltage falls below a threshold specific to the user’s range, the board will re-enter the user’s range.

The status of the input protection system can be read by the user application.

**2.3.3 ACV Zero Crossing Detection**

Accurate zero crossing detection is critical for the ACV RMS and frequency measurements. The following zero crossing modes are supported:

- Fixed zero level set by user
- (Max-Min)/2 value over a user-configurable number of samples
- Average DC value over the last period
- Average DC value over a user-configurable number of samples

**2.3.4 ACV True RMS Conversion**

The DNx-DMM-261 ACV meter is a True RMS voltmeter with AC-coupling.

The RMS voltage of an AC waveform equals the DC voltage that would deliver the same average power to a resistive load. The true RMS conversion used by the DNx-DMM-261 provides more accurate results compared to the simple averaging technique employed by less sophisticated meters. True RMS voltage is computed using the following formula:

$$V_{\text{trms}} = \sqrt{\frac{V_1^2 + V_2^2 + \dots + V_n^2}{n}}$$

where

- $V_{\text{trms}}$  is the true RMS voltage
- $n$  is the total number of samples
- $V_n$  is the instantaneous ADC voltage reading for sample  $n$

The total energy is calculated by integrating the instantaneous power over time (resistance is assumed to be equal to 1). Dividing the total energy by time gives the effective DC power, and taking the square root converts power back to voltage.

Measurement channels are available for acquiring the following ACV RMS measurements:

- True RMS AC Voltage is computed over the last detected AC period. If the Averaged RMS AC Voltage measurement channel is also specified in the channel list, the True RMS AC Voltage measurement channel will return the RMS value averaged over one period.
- Averaged RMS AC Voltage: This accumulate and divide mode computes true RMS voltage from data acquired since the previous read of the measurement channel. Each time the channel is read, the accumulated data is cleared and averaging restarts with new data.

**2.3.5 ACV Frequencies**

The DNx-DMM-261 can measure ACV signals between 20 Hz and 3 kHz. Frequency is calculated from the zero crossings described in Section 2.3.3.



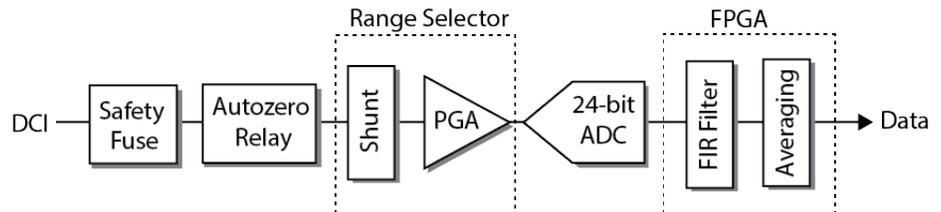
**2.3.6 ACV Peak Values**

Measurement channels are provided for reading the maximum and minimum values of the AC-coupled ACV waveform ( $V_{peak+}$  and  $V_{peak-}$ ). Peak measurements are filtered through a configurable moving average with a window size between 1 and 128 samples. Users can select a separate window size for AC voltage and AC current.

**2.4 DC Current (DCI) Measurement**

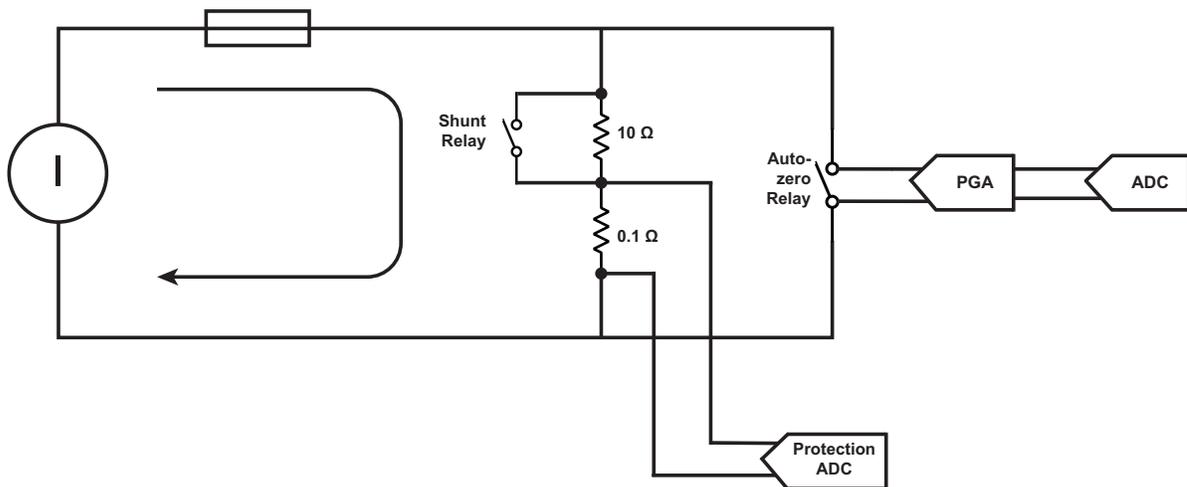
The DNx-DMM-261 can measure DC currents between 1.5 mA and 2 A.

**Figure 2-6** illustrates the main functional blocks in the DCI measurement path. The design centers around a high-speed (1.6 MS/sec) 24-bit simultaneous sampling auto-averaging ADC with a PGA front end. Note that DCI and ACI share the same hardware.



**Figure 2-6 Simplified Block Diagram for DC Current**

The circuit for a typical DCI measurement is shown in **Figure 2-7**. The input can be disconnected by a 2.5 A safety fuse or an auto-zero switch. The board directs the current through either a 10 Ω or 0.1 Ω shunt, measures the voltage across the shunt, and calculates the current value using Ohm's Law ( $I = V/R$ ).



**Figure 2-7 Connections for DC Current Measurement**

**2.4.1 DCI Range Selection** The DNx-DMM-261 supports the DCI ranges listed in **Table 2-3** along with the under-range, overrange, and clip point values for each range. The DNx-DMM-261 firmware will clip the current at the listed clip point value.

**Table 2-3 DCI Ranges ( $\pm$ )**

Range	Shunt	Under-range	Overrange	Clip Point
2 A	0.1 $\Omega$	1.2 A	n/a	2.4 A
1.5 A	0.1 $\Omega$	0.12 A	1.53 A	1.562 A
150 mA	0.1 $\Omega$	12 mA	180 mA	284 mA
15 mA	10 $\Omega$	1.2 mA	18 mA	31 mA
1.5 mA	10 $\Omega$	n/a	1.8 mA	2 mA

For a given DCI range, current is directed through the proper shunt resistor and the shunt voltage is amplified to utilize the full ADC scale.

Users can either select a range manually or configure the DCI channel for auto-ranging. Auto-ranging is useful for measuring a signal with an unknown or varying range. In auto-ranging mode, the DMM automatically switches the range up if the input is overrange or down if the input is under-range (**Table 2-3**). The board maximizes resolution by returning data from the smallest acceptable range. It also returns the range used for the measurement.

**2.4.2 DCI Input Protection** The Guardian current monitoring system is designed to protect the low current, 10  $\Omega$  shunt path. Current is continuously monitored by an independent ADC (separate from the one shown in **Figure 2-6**), even when the DMM is not actively making measurements.

If the average DC current exceeds 85 mA while in the 1.5 mA or 15 mA range, the board will switch over to the 2 A range (high current, 0.1  $\Omega$  shunt path). A single sample over 85 mA would not trip the protection.

After the protection trips, a board configured for auto-ranging will proceed as usual. In manual ranging mode, “auto-range” and “overrange” status flags are set to let the user know about the range change. Once the data falls within the limits of the user configured range, the board will re-enter the user’s range.

A 2.5 A fuse at the input guards against catastrophic overcurrent.

**2.4.3 DCI Auto-zero** Users can auto-zero DC current measurement channels to correct for any offsets in the measurement path. When the user calls the low-level API function `DqAdv261AutoZero()`, the auto-zero relay disconnects the external inputs from the DMM input circuitry. The board takes an average zero value reading over the specified duration and then reconnects the inputs. The new zero values are automatically subtracted from future measurements. Auto-zero can be enabled in the Framework API by calling `EnableAutoZero()`.



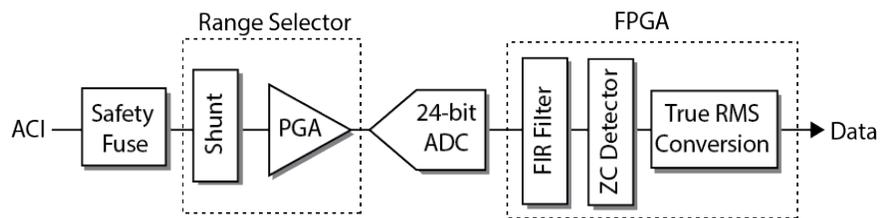
### 2.4.4 DCI Averaging

Samples from the ADC can be buffered and averaged to achieve higher resolution and accuracy. After the specified number of samples have accumulated in the memory stack, their average value is returned to the user and the stack is cleared. DCI supports the same sample averaging modes as DCV:

- Averaging OFF: Returns the calibrated DC current with no sample averaging.
- Averaging ON: This accumulate and divide mode averages all DCI samples acquired since the measurement channel was last read. This mode is ideal for applications where data is read infrequently.
- NPLC Averaging: Averages DCI samples over a configurable number of power line cycles. This mode is useful for eliminating AC power line noise, since a sine wave averages out to zero over a whole number of periods. DCV and DCI use the same NPLC configuration.

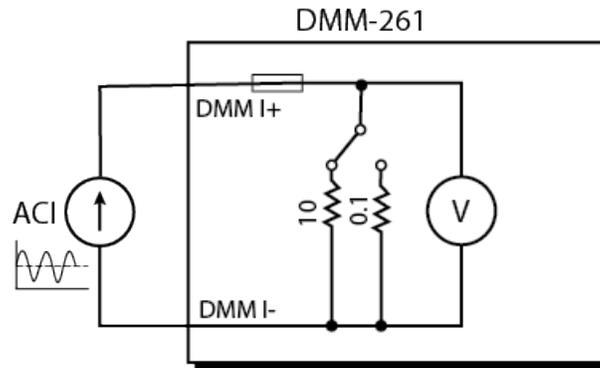
### 2.5 AC Current (ACI) Measurement

The DNx-DMM-261 can measure true RMS AC currents between 1.5 mArms and 2 Arms. **Figure 2-8** illustrates the main functional blocks in the ACI measurement path. The design centers around a high-speed (1.6 MS/sec) 24-bit simultaneous sampling auto-averaging ADC with a PGA front end. Note that the ACI subsystem shares hardware with the DCI subsystem.



**Figure 2-8 Simplified Block Diagram for AC Current**

The circuit for a typical ACI measurement is shown in **Figure 2-9**. The input is protected by a 2.5 A safety fuse as well as a software-controlled monitoring system (Section 2.5.2). The board directs the current through either a 10 Ω or 0.1 Ω shunt, measures the voltage across the shunt, and calculates the current value using Ohm's Law ( $I = V/R$ ). Unlike AC voltage, AC current is DC coupled to reduce settling time. The returned values are always relative to DC zero.



**Figure 2-9 Connections for AC Current Measurement**

**2.5.1 ACI Range Selection**

The DNx-DMM-261 supports the ACI ranges listed in **Table 2-4** along with the under-range, overrange, and clip point values for each range. The DNx-DMM-261 firmware will clip the current at the listed clip point value.

**Table 2-4 ACI Ranges (±)**

Range	Shunt	Under-range	Overrange	Clip Point
2 Arms	0.1 Ω	0.8 Arms	n/a	2.2 Arms
1 Arms	0.1 Ω	0.12 Arms	1.1 Arms	1.562 Arms
150 mArms	0.1 Ω	12 mArms	180 mArms	284 mArms
15 mArms	10 Ω	1.2 mArms	18 mArms	30.1 mArms
1.5 mArms	10 Ω	n/a	1.8 mArms	281 mArms

For a given ACI range, current is directed through the proper shunt resistor and the shunt voltage is amplified to utilize the full ADC scale.

Users can either select a range manually or configure the ACI measurement channel for auto-ranging. Auto-ranging is useful for measuring a signal with an unknown or varying range. In auto-ranging mode, the DMM automatically switches the range up if the input is over-range or down if the input is under-range (**Table 2-4**). The board maximizes resolution by returning data from the smallest acceptable range. It also returns the range used for the measurement.

**2.5.2 ACI Input Protection**

The Guardian current monitoring system is designed to protect the low current, 10 Ω shunt path. Current is continuously monitored by an independent ADC (separate from the one shown in **Figure 2-8**), even when the DMM is not actively making measurements.

Protection will trip if one of the following conditions occur while in the 1.5 mArms or 15 mArms range:

- AC current exceeds 60 mArms
- Average DC current exceeds 85 mA



When protection trips, the board will switch over to the 2 Arms range (high current, 0.1  $\Omega$  shunt path). Auto-ranging, if configured, will proceed as usual. In manual ranging mode, “auto-range” and “overrange” status flags are set to let the user know about the range change. Once the data falls within the limits of the user configured range, the board will re-enter the user’s range.

A 2.5 A fuse at the input guards against catastrophic over-current.

**2.5.3 ACI Zero Crossing Detection**

Accurate zero crossing detection is critical for the ACI RMS and frequency measurements. The following zero crossing modes are supported:

- Fixed zero level set by user
- (Max-Min)/2 value over a user-configurable number of samples
- Average DC value over the last period
- Average DC value over a user-configurable number of samples

**2.5.4 ACI True RMS Conversion**

The RMS current of an AC waveform equals the DC current that would deliver the same average power to a resistive load. The DNx-DMM-261 ammeter computes the true RMS voltage across the shunt as described in Section 2.3.4. RMS voltage is then divided by the shunt resistance to obtain the RMS current.

Measurement channels are available for acquiring the following ACI RMS measurements:

- True RMS AC Current is computed over the last detected AC period. If the Averaged RMS AC Current measurement channel is also specified in the channel list, the True RMS AC Current measurement channel will return the RMS value averaged over one period.
- Averaged RMS AC Current: This accumulate and divide mode computes true RMS current from data acquired since the previous read of the measurement channel. Each time the measurement channel is read, the accumulated data is cleared and averaging restarts with new data.

**2.5.5 ACI Frequencies**

The DNx-DMM-261 can measure ACI signals between 20 Hz and 3 kHz. Frequency is calculated from the zero crossings described in Section 2.5.3.

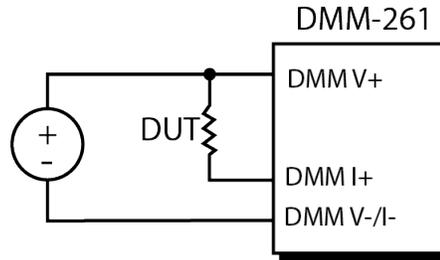
**2.5.6 ACI Peak Values**

Measurement channels are provided for reading the maximum and minimum values of the DC-coupled ACI waveform ( $I_{peak+}$  and  $I_{peak-}$ ). Peak measurements are filtered through a configurable moving average with window size between 1 and 128 samples. Users can select a separate window size for AC voltage and AC current.



## 2.6 Power Measurement

The DNx-DMM-261 can measure DC power up to 900 watts and AC apparent power up to 600 watts. Connect the device under test (DUT) to the DMM as shown in **Figure 2-10**. Voltage and current channels share the same ground so only three wires to the DMM are needed.



**Figure 2-10 Connections for Power Measurement**

To determine the power, measure voltage and current on separate channels and multiply the results.

$$\text{DC Power} = V_{\text{DC}} \times I_{\text{DC}}$$

$$\text{AC Power} = V_{\text{trms}} \times I_{\text{trms}}$$

where

- $V_{\text{DC}}$  is the DC voltage (Section 2.2)
- $I_{\text{DC}}$  is the DC current (Section 2.4)
- $V_{\text{trms}}$  is the true RMS voltage (Section 2.3)
- $I_{\text{trms}}$  is the true RMS current (Section 2.5)

Voltage and current measurement channels are described in their respective sections.

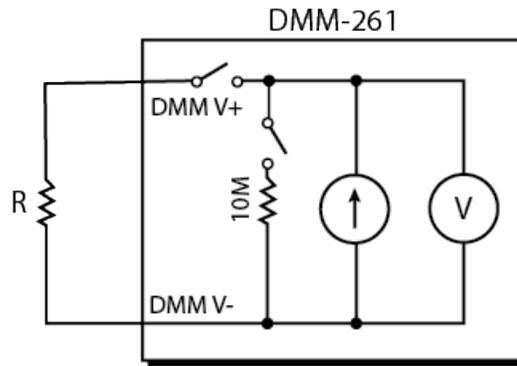
## 2.7 Resistance Measurement

The DNx-DMM-261 supports 2-wire measurements from 10  $\Omega$  to 100 M $\Omega$  or 4-wire measurements from 10  $\Omega$  to 10 M $\Omega$ . The 2-wire method is typically used when the test lead resistance is negligible relative to the DUT. For low-resistance devices, the 4-wire method is preferred because it eliminates the relatively large errors introduced by the leads.

In both modes, the DNx-DMM-261 supplies a constant excitation current to the DUT, measures the resultant voltage drop across the DUT, and calculates its resistance using  $R=V/I$ .

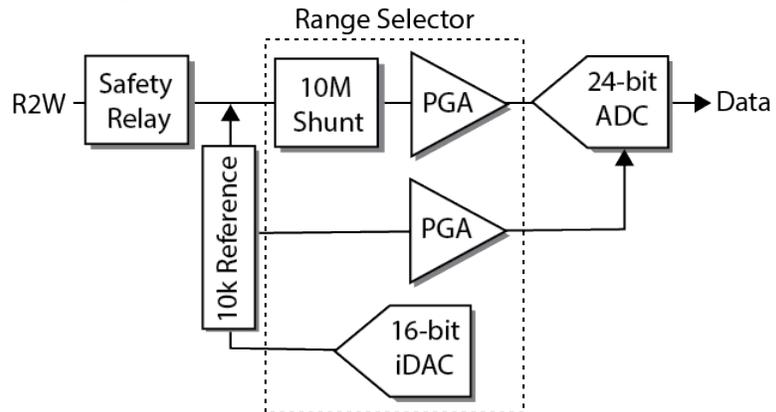
### 2.7.1 2-Wire Resistance (R2W)

As shown in **Figure 2-11**, a 2-wire resistance measurement sources current and measures voltage via the same two terminals (V+ and V-). This simple wiring scheme is typically sufficient for measuring medium to high loads (>1 k $\Omega$ ), depending on the accuracy desired.



**Figure 2-11 Connections for 2-Wire Resistance Measurement**

**Figure 2-12** illustrates the main functional blocks involved in the 2-wire measurement. A 16-bit DAC generates an excitation current between 1  $\mu$ A and 1 mA, which passes through a high precision 10k  $\Omega$  reference resistor and through the DUT. The voltage across the DUT is amplified and read by the 24-bit ADC. The voltage across the reference resistor is also amplified and serves an ultra stable reference to the ADC. By adjusting the excitation current and the gains on the two PGAs, the DMM can directly measure 10  $\Omega$  to 10 M $\Omega$  resistances using the full ADC scale.



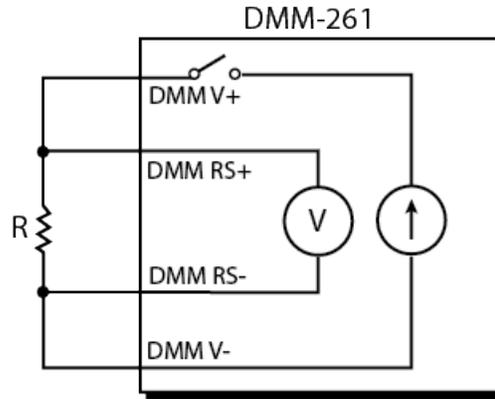
**Figure 2-12 Simplified Block Diagram for 2-Wire Resistance**

When using the 100 M $\Omega$  range, a 10 M $\Omega$  2 ppm shunt resistor is switched in parallel with the DUT (see **Figure 2-11**). The firmware automatically accounts for the new equivalent resistance seen by the ADC and returns the actual DUT resistance.

### 2.7.2 4-Wire Resistance (R4W)

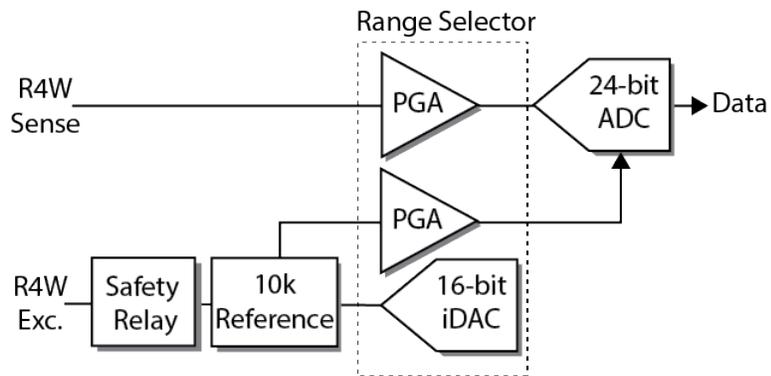
As shown in **Figure 2-13**, a 4-wire resistance measurement sources current through the V+/V- terminals and measures voltage across the RS+/RS- (“Resistance Sense”) terminals. The sense lines are connected as close to the resistor terminals as possible. Therefore, the measured voltage no longer includes the voltages across the current source leads. In addition, the voltmeter’s high impedance prevents current flowing through the sense leads, so the only voltage drop is across the DUT itself.

Because the measurement is independent of lead resistance, 4-wire mode is ideal for low-resistance DUTs (<1 k $\Omega$ ) or situations where the lead resistance varies (e.g., when inputs are connected through a multiplexer with varying path lengths).



**Figure 2-13 Connections for 4-Wire Resistance Measurement**

**Figure 2-14** illustrates the main functional blocks involved in the 4-wire measurement. A 16-bit DAC generates an excitation current between 1  $\mu$ A and 1 mA, which passes through a high precision 10 k $\Omega$  reference resistor and through the DUT via the excitation leads. The voltage on the sense leads is amplified and read by the 24-bit ADC. The voltage across the reference resistor is also amplified and serves an ultra stable reference to the ADC. By adjusting the excitation current and the gains on the two PGAs, the DMM can directly measure 10  $\Omega$  to 10 M $\Omega$  resistances using the full ADC scale. Four-wire measurements use the same circuitry as 2-wire measurements, except it lacks the 10 M $\Omega$  shunt resistor and therefore does not support the 100 M $\Omega$  range.



**Figure 2-14 Simplified Block Diagram for 4-Wire Resistance**

### 2.7.3 Resistance Ranges

The DNx-DMM-261 provides the resistance ranges listed in **Table 2-5**:

**Table 2-5 Resistance Ranges**

Range	Under-range	Overrange	Over-voltage
10 Ω	n/a	14 Ω	±13 V
100 Ω	5 Ω	113 Ω	±13 V
1 kΩ	50 Ω	1.25 kΩ	±13 V
10 kΩ	0.5 kΩ	13.7 kΩ	±13 V
100 kΩ	5 kΩ	116 kΩ	±13 V
1 MΩ	50 kΩ	1.28 MΩ	±13 V
10 MΩ	0.5 MΩ	14.1 MΩ	±13 V
100 MΩ (2-wire measurements only)	5 MΩ	n/a	±13.5 V

Users can either select a range manually or configure the resistance measurement channel for auto-ranging. Auto-ranging is useful for measuring a signal with an unknown or varying range. In auto-ranging mode, the DMM automatically switches the range up if the input is overrange or down if the input is under-range (**Table 2-5**). The board maximizes resolution by returning data from the smallest acceptable range. It also returns the range used for the measurement.

If no resistance continuity is detected, a flag will be set in the returned data.

### 2.7.4 Resistance Input Protection

Resistance measurements are normally performed with the circuit power turned off to avoid damaging the circuit or the DMM. However, the DNx-DMM-261 includes a front-end safety relay in the event that voltage is accidentally applied to the inputs in resistance mode.

When the voltage on the ADC exceeds the over-voltage limit (**Table 2-5**), the firmware will open the safety relay and reconnect in DCV mode in the 300 V range. The firmware waits until the input voltage falls below 1 V to switch back to resistance mode. If the input voltage exceeds 350 V during this waiting period, the input will once again be disconnected and the board will need to be reconfigured by the user.



## 2.8 Guardian Diagnostics

Two independent Guardian ADCs are used to monitor board health: a 16-bit Voltage Monitor and a 12-bit Current Monitor.

### 2.8.1 Voltage Monitor

The Guardian Voltage Monitor can read:

- Voltage of internal -17.5 V supply
- Voltage of internal +17.5 V supply
- Voltage of internal -5.5 V supply
- Voltage of internal +5.5 V supply
- On-board temperature

### 2.8.2 Current Monitor

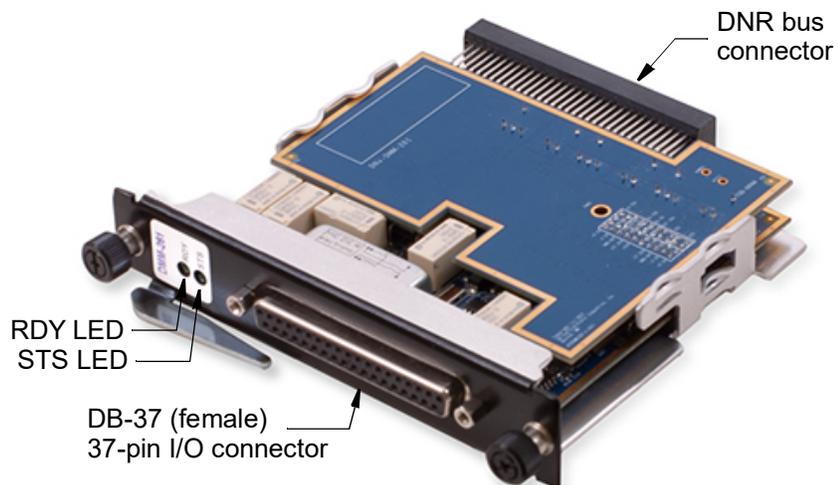
The Guardian Current Monitor is always on and triggers input protection features when over-current conditions occur (see Section 2.4.2 and Section 2.5.2). Users can read back the following data:

- DC current
- AC RMS current
- Minimum value of AC current ( $I_{\text{peak}^-}$ )
- Maximum value of AC current ( $I_{\text{peak}^+}$ )

**NOTE:** The Guardian Current Monitor is intended for diagnostics only, not for making DMM current measurements. The Guardian channels are not calibrated.

## 2.9 Indicators and Connectors

Figure 2-15 shows the locations of the LEDs and connectors on the DNx-DMM-261. The LED indicators are described in Table 2-6.



**Figure 2-15 Photo of DNR-DMM-261 Board**

**Table 2-6 DNx-DMM-261 LED Indicators**

LED Name	Description
RDY	Board is powered up and operational
STS	<p><b>DCV/ACV/Resistance</b> - ON when safety relay is closed, OFF when protection is tripped in the maximum input range. The safety relay will be disconnected.</p> <p><b>DCI</b> - ON when current measurement is below 2.2 A, otherwise OFF</p> <p><b>ACI</b> - ON when RMS measurement is below 2.2 A, otherwise OFF</p>

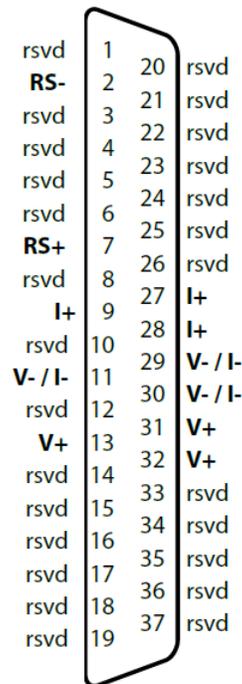
## 2.10 Pinout

**Figure 2-16** illustrates the pin configuration for the DNx-DMM-261 board. External connections are made through a standard DB-37 female connector. Signal connections are shown in **Table 2-7**. If using the DNx-DMM-261 with DNx-MUX-461 multiplexer boards, DMM connections are made internally within the chassis.

### **No Hot Swapping!**



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.



**Figure 2-16 Pinout Diagram for the DNx-DMM-261**



**Table 2-7 Signal Connections for the DNx-DMM-261**

Measurement Type	Connection
AC/DC Voltage	V+ positive terminal V- negative terminal
AC/DC Current	I+ positive terminal I- negative input terminal
2-wire Resistance	V+ positive V- negative
4-wire Resistance	V+ positive* excitation current V- negative* excitation current  RS+ Resistance sense RS- Resistance sense return

\* polarity shown to indicate current flow direction. This may be ignored for pure resistance measurements, but if semiconductors are in circuit, current direction often matters (e.g., if there is a series diode).

**NOTE 1:** Multiple connections are shown for V and I pins. For standard measurements only one pin typically needs connection, though on two wire resistance measurements better accuracy will be achieved by using all three pins.

**NOTE 2:** The V-/I- pins are internally connected on the PCB, but for the most accurate current measurement, pin 29 should be used for I- input for current under 1 A. Current measurements above 1 A should use pins 11 and 30 to reduce current load in the cable.

**NOTE 3:** Pins on the connector are only rated to 2 amps. If you are using current inputs that exceed 1 amp, it is recommended you use all three I+ and I- current input pins.

**Unused Pins**



Unused pins may be left open/disconnected. Do not connect anything to the pins marked as “rsvd”.

## 2.11 Multiplexer Boards for the DMM-261

The DNx-MUX-461 Series multiplexer boards are high-speed switch interfaces designed to provide additional input channels to the DNx-DMM-261. Up to five MUX boards may be daisy chained within a single Cube/RACK chassis, providing additional input channels as follows:

- up to 130 two-wire or 65 four-wire input channels using the DNx-MUX-461.
- up to 120 two-wire or 60 four-wire input channels using the DNx-MUX-461-350 high voltage multiplexer board.

Note that the number of available input channels can vary depending on your configuration. Inputs from a single MUX board may consist of both two and four-wire input channels. Your configuration may also include a mixture of DNx-MUX-461 and DNx-MUX-461-350 MUX boards.

Refer to the “DNx-MUX-461 User Manual” for specifications, relay diagrams, and additional information.

A DMM+MUX system ships with all inter-board connections made inside the Cube/RACK, so the only connections you need to make are to the various input channels on the MUX boards. To install or remove MUX boards in the field, refer to:

- Connecting DMM and MUX Boards within a Cube (Section 2.11.1)
- Connecting DMM and MUX Boards within a Rack (Section 2.11.2)

### 2.11.1 Connecting DMM and MUX Boards within a Cube

In a Cube chassis, DNA-DMM-261 and DNA-MUX-461 boards are internally connected by the JPOW1 connectors on the PCBs. Each board plugs into the one above and below it in the stack. No cable is required.

**NOTE:** If you use the internal DMM-to-MUX connection, you cannot use a DNA-PC-91x Series board to provide internal power to other I/O boards in the stack. This is because the DMM-261, MUX-461, and DNA-PC-91x boards use the same JPOW1 connection pins.

The API assumes that MUX-461 boards are stacked directly below the DMM-261. For example, if your DMM is located in I/O Slot 2 and you have three MUX boards, install the MUX boards in Slots 3, 4, and 5.

Each Cube I/O board includes a jumper block that identifies the physical position of the I/O board in the stack. If you change the order of your I/O boards in your Cube or install a new board, you may need to adjust the corresponding hardware jumpers.

Refer to the *PowerDNA Field Installation Guide* for more information about installing I/O boards and setting jumper positions in Cubes.

### 2.11.2 Connecting DMM and MUX Boards within a Rack

In a RACKTangle or HalfRACK chassis, DNR-DMM-261 and DNR-MUX-461 Series boards are internally connected by a DNR-CBL-461-x cable, where x is the number of MUX boards (1, 2, 3, 4, or 5). As shown in **Figure 2-17**, the cable plugs into each board in a daisy-chain configuration.





**Figure 2-17 DNR-DMM-261 (left) connected to five DNR-MUX-461 Boards**

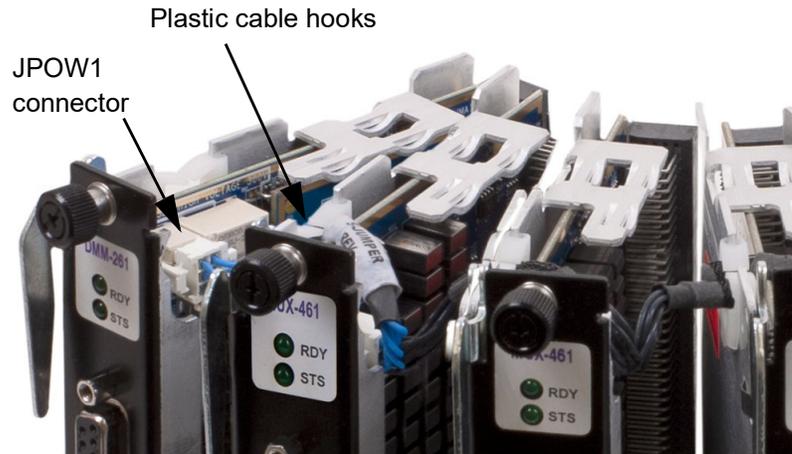
To install DMM and MUX modules into a RACKTangle or HalfRACK system, do the following:

- STEP 1:** Turn off power to the Rack.
- STEP 2:** Attach a grounding strap to your wrist, and follow normal ESD procedures.
- STEP 3:** Remove any external cables from the DMM and MUX modules.
- STEP 4:** With the modules outside of the enclosure, orient the modules vertically so that the LEDs are above the D-Sub connectors (**Figure 2-17**). Arrange the modules such that the DMM-261 is positioned on the far left.



The API assumes that MUX-461 boards are located directly after the DMM-261 in the chassis. For example, if your DMM will be placed in I/O Slot 2 and you have three MUX boards, place the MUX boards in Slots 3, 4, and 5.

Plug one end of the DNR-CBL-461-x cable into the JPOW1 connector on the DMM-261 (refer to **Figure 2-18**).



**Figure 2-18 Installing DMM-MUX Cable**

**STEP 5:** Gently push the DNR-CBL-461-x cable through the flexible plastic hooks on the first MUX-461 module, and plug the cable into the JPOW1 connector on the MUX-461.

The connector is asymmetrical and only fits in one orientation. The fit will be tight to ensure that the cable does not move around.

**STEP 6:** Repeat Step 5 for the other MUX-461 modules.

**STEP 7:** Insert the module assembly into the Rack enclosure:

- a. Line up the modules with the I/O slot guides and slide the modules into the chassis.  
 We suggest flipping the enclosure face-up and lowering the modules from above until partially inserted. Then flip the enclosure back on its side and push the modules all the way in.
- b. Use the insertion levers to seat each module into the backplane connector.
- c. Using a torque screwdriver set to 5 in-lb, screw in the thumbscrews until the torque screwdriver clicks.

**STEP 8:** Reattach the field wiring and restore power to the system.

## Chapter 3 PowerDNA Explorer

This chapter provides the following information about exploring the DNx-DMM-261 with the PowerDNA Explorer application.

- Introduction (Section 3.1)
- Read DMM Data (Section 3.2)
- Configuration with MUX-461 Boards (Section 3.3)

### 3.1 Introduction

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 can be launched from the Windows startup menu:

**Start » All Programs » UEI » PowerDNA Explorer**

The DNx-DMM-261 is supported in PowerDNA version 5.2+.

When using PowerDNA Explorer to read the DNx-DMM-261, ensure that no other applications are communicating with the IOM (e.g., via C code or LabVIEW). You may have to restart PowerDNA Explorer to re-establish communication with the board.

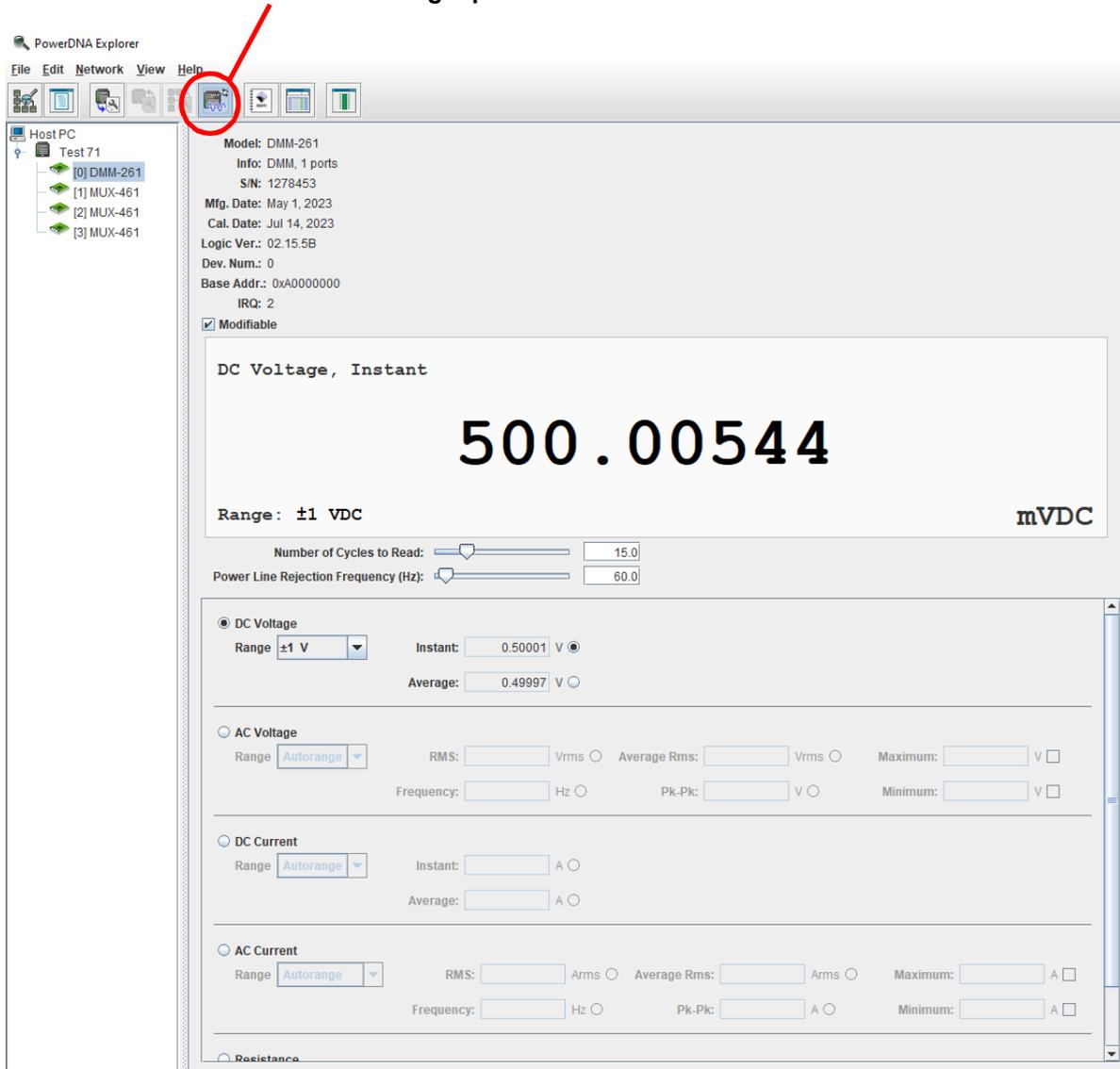
**NOTE:** PowerDNA Explorer only supports basic DNx-DMM-261 functionality. To access additional features, refer to Chapter 4, “Programming with the High-level API” or Chapter 5, “Programming with the Low-level API”.



### 3.2 Read DMM Data

As shown in Figure 3-1, PowerDNA Explorer for the DNx-DMM-261 contains a read-only device information display, a slider for NPLC configuration, and a measurement mode configuration panel

#### Start Reading Input Data



**Figure 3-1 PowerDNA Explorer for DNx-DMM-261**

To read DMM data:

1. Select a DMM measurement type (DCV, ACV, DCI, ACI, or Resistance) in the measurement mode configuration panel.
2. Select a **Range** from the drop-down menu.
3. Select a measurement mode to display in the large read-only display. Data for all other measurement modes will be displayed next to the measurement mode names.
4. Click the **Start Reading Input Data** button. Data is continuously updated while the read button is active.



### 3.2.1 DC Voltage and Current

The following DCV and DCI measurement data are available in PowerDNA Explorer:

- **Instant DC Voltage/Current:** Calibrated DC voltage or current with no averaging.
- **Average:** DC voltage or current averaged over the number of samples determined by NPLC.

### 3.2.2 AC Voltage and Current

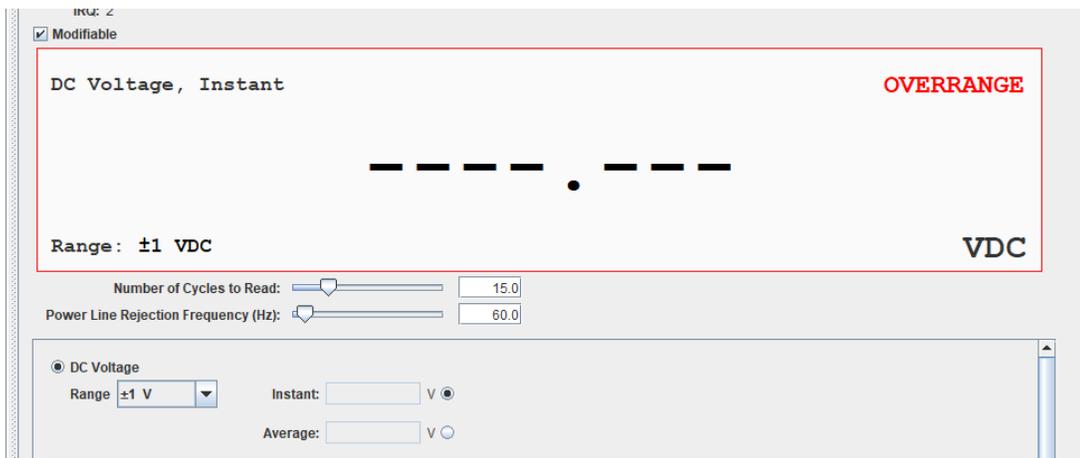
The following ACV and ACI measurement data are available in PowerDNA Explorer:

- **RMS:** True RMS value calculated over last detected AC period.
- **Pk-Pk:** Peak to peak amplitude. This is the difference between the minimum and maximum voltage or current.
- **Frequency:** Last detected AC frequency
- **Minimum:** Minimum peak value averaged over the last 2 readings.
- **Maximum:** Maximum peak value averaged over the last 2 readings.

### 3.2.3 Overrange Error

If you see an “Overrange” error message (**Figure 3-2**), it means that the input exceeds the user-configured range. When overrange occurs, the DNx-DMM-261 automatically disconnects the input and reconnects in a safe range.

To resolve the error, either select a higher range or wait for the input to re-enter the existing range. Once data falls within the user-configured range, the board will automatically reconfigure to that range.



**Figure 3-2 Example of Overrange Error**

### 3.2.4 Protection Tripped Indicator

If an input to the DNx-DMM-261 causes protection to be tripped, the DNx-DMM-261 firmware will switch to a safe range and a “PROTECTED” indicator will be displayed as shown in **Figure 3-3**. The measured value will be displayed in the channel configuration panel.



**Figure 3-3 Protection Tripped Indicator**

### 3.2.5 Resistance Channel

You can configure the resistance channel for either 2-wire or 4-wire measurements. Note that the 100 MΩ range can only be selected for 2-wire measurements.

### 3.2.6 Auto-ranging

The device information display will indicate if the selected measurement has been configured for auto-ranging, as shown in **Figure 3-4**.

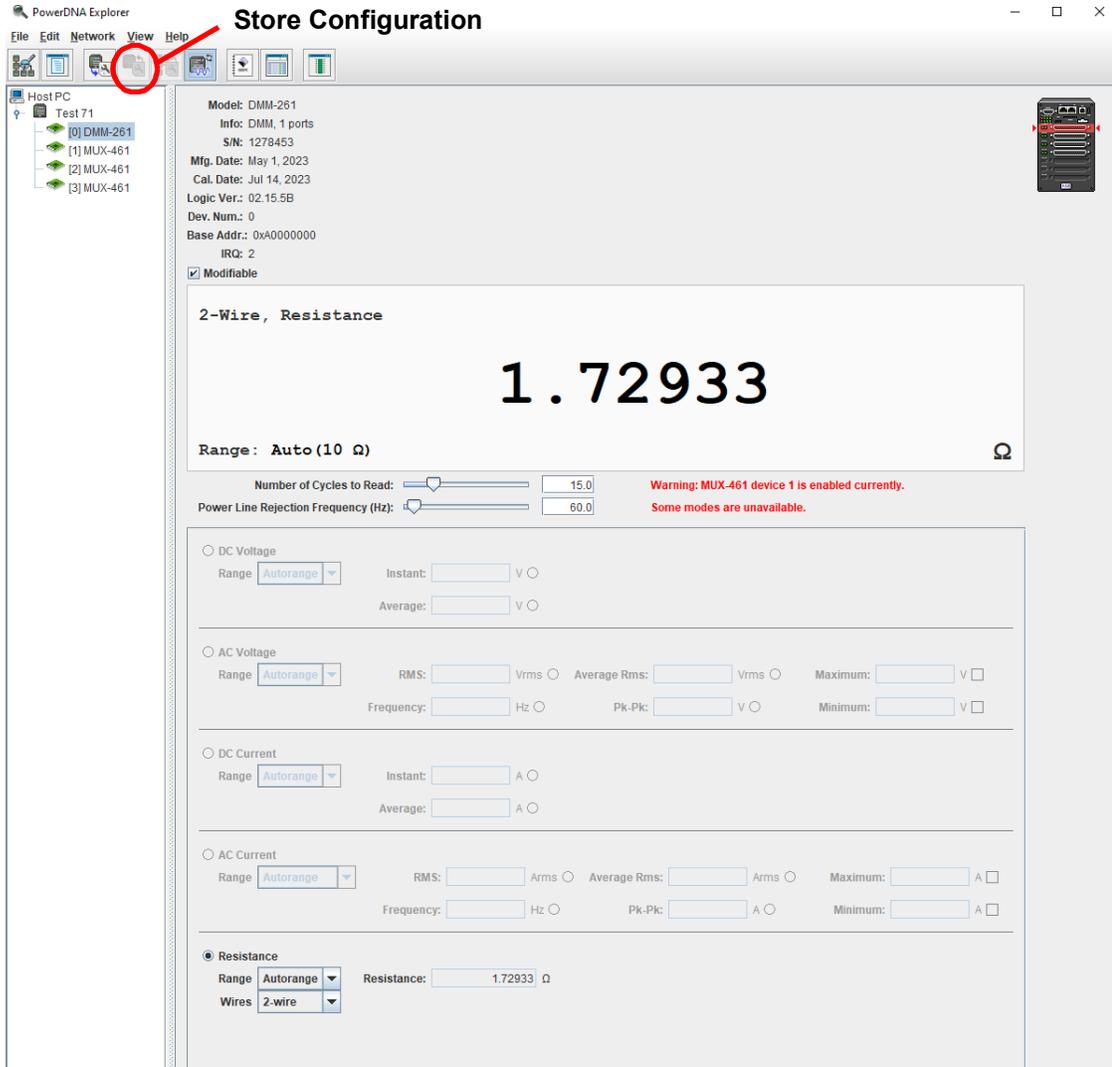


**Figure 3-4 Auto-ranging Indicator**



### 3.3 Configuration with MUX-461 Boards

Figure 3-5 shows PowerDNA Explorer when a system has been configured with a DNx-DMM-261 and three MUX-461 boards. Select a MUX-461 device in PowerDNA Explorer and click the “Store Configuration” button to enable the selected MUX board.



**Figure 3-5 Configuration with DMM-261 and MUX-461 Boards**



# Chapter 4 Programming with the High-level API

This chapter provides the following information about programming the DNx-DMM-261 using the UeiDaq Framework API:

- About the High-level API (Section 4.1)
- Example Code (Section 4.2)
- DMMChannel Methods (4.3)
- Create a Session (Section 4.4)
- Resource Strings (4.5)
- Configure DMM Channels (Section 4.6)
- Configure the Timing (Section 4.7)
- Start the Session (Section 4.8)
- Read Data (Section 4.9)
- Stop the Session (Section 4.10)
- Using the DMM-261 with the MUX-461 (Section 4.11)

## 4.1 About the High-level API

The UeiDaq Framework API is object oriented and its objects can be manipulated in the same manner from different development environments, such as Visual C++, MATLAB, LabVIEW, and more. The Framework is supported in Windows 7 and up. It is generally simpler to use compared to the low-level API, and it includes a generic simulation device to assist in software development. Therefore, we recommend that Windows users use the Framework API unless unconventional functionality is required. Users programming in Linux or a real-time operating system should instead use the low-level API (Chapter 5).

For more detail regarding the Framework’s architecture, please see the “*UeiDaq Framework User Manual*” located under:

**Start » All Programs » UEI**

For information on the Framework’s classes, structures, and constants, please see the “*UeiDaq Framework Reference Manual*” located under:

**Start » All Programs » UEI**

## 4.2 Example Code

The UeiDaq Framework is bundled with examples for supported programming languages. The example code is located in language-specific folders under:

**C:\Program Files (x86)\UEI\Framework**

For example:

**Start » All Programs » UEI » Visual C++ Examples**

Unlike the low-level examples, Framework examples are board-agnostic. For instance, the “DMM” example applies to all UEI digital multimeter boards including the DNx-DMM-261.



Each high-level example follows the same basic structure:

1. Create a session.
2. Configure the session for a particular device and subsystem.
3. Configure the timing.
4. Start the session.
5. Read or write data.
6. Stop the session.

The remainder of this chapter is intended as a supplement to the example code. Examples are presented using the C++ API, but the concepts are the same no matter which programming language you use. The “UeiDaq Framework User Manual” provides additional information about programming in other languages.

### 4.3 DMMChannel Methods

As shown throughout Section 4.6, the `CUeiDMMChannel` class is used to configure and manage DMM measurements. The `CUeiDMMChannel` class implements the methods listed in **Table 4-1**. See the “UeiDaq Framework Reference Manual” for more details including information on parent classes.

**Table 4-1 CUeiDMMChannel Class Methods**

Get Methods	Set Methods
<code>GetMeasurementMode</code>	<code>SetMeasurementMode</code>
<code>GetZeroCrossingNumberOfSamples</code>	<code>SetZeroCrossingNumberOfSamples</code>
<code>GetPowerLineFrequency</code>	<code>SetPowerLineFrequency</code>
<code>GetNumberOfPowerLineCycles</code>	<code>SetNumberOfPowerLineCycles</code>
<code>GetDisconnectTimeout</code>	<code>SetDisconnectTimeout</code>
<code>IsAutoRangeEnabled</code>	<code>EnableAutoRange</code>
<code>GetMuxDeviceCount</code>	<code>SetMuxDeviceCount</code>

### 4.4 Create a Session

The session object manages all communications with the DNx-DMM-261. Therefore, the first step is always to create a new session.

```
//create a session object
CUeiSession dmmSession;
```

### 4.5 Resource Strings

When calling methods such as `CreateDMMChannel()`, the Framework uses a resource string to link the session to the hardware. The resource string syntax is similar to a web URL; it should not have any spaces and is case insensitive. The resource string is formatted as:

“<device class>://<IP address>/<device number>/<subsystem><channel list>”

- <device class> – Set the device class to `pdna`.
- <IP address> – IP address of the IOM.



- *<device number>* – position of the DNx-DMM-261 within the chassis, relative to the other I/O boards.
- *<subsystem>* – the DNx-DMM-261 uses the `Ai` subsystem.
- *<channel list>* – all V, I, and R measurements on the DNx-DMM-261 are through channel 0 of the `Ai` subsystem, i.e., `Ai0`.

**Example**

The following resource string is used to access a DNx-DMM-261 at IP address 192.168.100.2, `DEVN1`:

```
"pdna://192.168.100.2/Dev1/Ai0"
```

**4.6 Configure DMM Channels**

You can configure the session for DMM measurements using the `CreateDMMChannel()` method. This section covers how to add and configure DMM channels for the following types of measurements or functionality:

- DCV and DCI (Section 4.6.1)
- ACV and ACI (Section 4.6.2)
- Resistance (Section 4.6.3)
- Guardian Diagnostics (Section 4.6.5)
- Time Stamp (Section 4.6.6)

The `CreateDMMChannel()` method creates and adds one or more analog input channels to the session's channel list.

```
UeiDaqAPI CUeiDMMChannel *
UeiDaq::CUeiSession::CreateDMMChannel(std::string resource, f64 range,
tUeiDMMMeasurementMode measurementMode);
```

The board will cycle through all programmed channels and return the data in a single data stream. When `CreateDMMChannel()` is called, the `measurementMode` argument is passed as an argument. Within a session, all members of the channel list must be of the same measurement mode type, e.g., NPLC averaged DC voltage, calibrated DC voltage, etc. Mixing measurement mode types may cause invalid data. The measurement mode constants used in UEI Framework applications are defined in `UeiConstants.h`.

**4.6.1 DC Measurement**

This section provides guidance for using the Framework API to configure DC voltage and current measurement.

**4.6.1.1 Configure DMM Channel for DCV**

The following example configures a DMM channel for DC voltage measurement, sets its range, and obtains a pointer to the channel.

```
//Add a DMM channel for DCV measurement in the 10 V range.
CUeiDMMChannel* dcv = dmmSession.CreateDMMChannel(
    "pdna://192.168.100.2/Dev0/Ai0", 10, UeiDMMModeCalibratedDCVoltage);
```



The following DC voltage measurement modes are available:

- `UeiDMMModeCalibratedDCVoltage` – Calibrated DC voltage with no averaging.
- `UeiDMMModeAveragedNplcDCVoltage` – DCV averaged over N power line cycles. See Section 4.6.1.4 for configuration.

See **Table 2-1** for a list of available DCV ranges. If you pass in an unsupported range value, the board defaults to the next available range. For example, if you request a range of 200 V, the board will use the 300 V range.

#### 4.6.1.2 Configure DMM Channel for DCI

The following example configures a DMM channel for DC current measurement, sets its range, and obtains a pointer to the channel.

```
//Add a DMM channel for DCI measurement in the 150 mA range.
CUeiDMMChannel* dci = dmmSession.CreateDMMChannel(
    "pdna://192.168.100.2/Dev0/Ai0", 0.15, UeiDMMModeCalibratedDCCurrent);
```

The following DC current measurement modes are available:

- `UeiDMMModeCalibratedDCCurrent` – Calibrated DCI with no averaging.
- `UeiDMMModeAveragedNplcDCCurrent` – DCI averaged over N power line cycles (see Section 4.6.1.4 for configuration).

See **Table 2-3** for a list of available DCI ranges. If you pass in an unsupported range value, the board defaults to the next available range. For example, if you request a range of 1.0 A, the board will use the 1.5 A range.

#### 4.6.1.3 Auto-Zero

DMM channels configured for DCV and DCI measurements may be independently auto-zeroed. If enabled, the channel will auto-zero for 3 seconds at the start of the session.

```
//Auto-zero the DCV channel when the session starts.
dcv->EnableAutoZero(TRUE);
```

#### 4.6.1.4 Configure NPLC

NPLC settings are used in the `UeiDMMModeAveragedNplcDCVoltage` and `UeiDMMModeAveragedNplcDCCurrent` modes to average out AC noise. You can configure the AC frequency and the number of cycles averaged per read:

```
//Set power line frequency to 60 Hz.
//Average DC samples over 120 AC cycles.
dcv->SetPowerLineFrequency(60);
dcv->SetNumberOfPowerLineCycles(120);
```

In DC modes, the maximum read duration is 15 seconds (900 cycles @ 60 Hz or 750 cycles @ 50 Hz).



All DMM channels share the same NPLC registers, so if you set NPLC on multiple channels, the latest configuration will apply to all DMM channels. New data will not be ready before the end of the time, calculated in seconds, needed to complete the read cycle (number of cycles divided by line frequency).

#### 4.6.2 AC Measurement

This section provides guidance for using the Framework API to configure AC voltage and current measurement.

##### 4.6.2.1 Add DMM Channel for ACV

The following example adds a DMM channel configured to measure AC voltage, sets its range, and obtains a pointer to the channel.

```
//Add an ACV channel in the 10 Vrms range.
CUEiDMMChannel* acv = dmmSession.CreatedMMChannel(
    "pdna://192.168.100.2/Dev0/Ai0", 10, UeiDMMModeRMSACVoltage);
```

The following AC voltage measurement modes are available:

- `UeiDMMModeRMSACVoltage` – True RMS voltage (AC-coupled) computed over the last detected AC period.
- `UeiDMMModeAveragedRMSACVoltage` – True RMS voltage computed over all samples accumulated since the previous read.
- `UeiDMMModeMinACPeakVoltage` – Minimum peak AC voltage, uses a moving average configured in Section 4.6.2.4.
- `UeiDMMModeMaxACPeakVoltage` – Maximum peak AC voltage, uses a moving average configured in Section 4.6.2.4.
- `UeiDMMModeFrequencyACVoltage` – Last detected ACV frequency. Range is ignored in this mode.

See **Table 2-2** for a list of available ACV ranges. If you pass in an unsupported range value, the board defaults to the next available range. For example, if you request a range of 1.0 Vrms, the board will instead use the 1.5 Vrms range.

##### 4.6.2.2 Add DMM Channel for ACI

The following example adds a DMM channel configured to measure AC current, sets its range, and obtains a pointer to the channel.

```
//Add a DMM channel to measure ACI in the 150 mArms range.
CUEiDMMChannel* acv = dmmSession.CreatedMMChannel(
    "pdna://192.168.100.2/Dev0/Ai0", 0.15, UeiDMMModeRMSACCurrent);
```

The following AC current measurement modes are available:

- `UeiDMMModeRMSACCurrent` – True RMS current (DC-coupled) computed over the last detected AC period.
- `UeiDMMModeAveragedRMSACCurrent` – True RMS current computed over all samples accumulated since the previous read.
- `UeiDMMModeMinACPeakCurrent` – Minimum peak AC current, subject to a moving average configured in Section 4.6.2.4.



- `UeiDMMModeMaxACPeakCurrent` – Maximum peak AC current, subject to a moving average configured in Section 4.6.2.4.
- `UeiDMMModeFrequencyACCurrent` – Last detected ACI frequency. Range is ignored in this mode.

See **Table 2-4** for a list of available ACI ranges. If you pass in an unsupported range value, the board defaults to the next available range. For example, if you request a range of 100 mArms, the board will instead use the 150 mArms range.

#### 4.6.2.3 Configure Number of Read Cycles

The DNx-DMM-261 computes the true RMS value over multiple periods of the AC input signal. To configure the number of read cycles:

```
//Compute RMS over 29 periods.
acv->SetNumberOfPowerLineCycles(29);
```

The maximum NPLC for AC measurements is 29 at 60 Hz or 24 at 50 Hz. The minimum is 1. All channels share the same NPLC registers, so if you set NPLC on multiple channels, the latest configuration will apply to all channels. New data will not be ready before the end of the time, calculated in seconds, needed to complete the read cycle (number of cycles divided by line frequency).

#### 4.6.2.4 Configure AC Peak Averaging

The DNx-DMM-261 uses a moving average when reading minimum and maximum peak values. The window size can be set to 1, 2, 4, 8, 16, 32, 64, or 128.

```
//Average 64 min/max values per read.
acv->SetMovingAverageWindowSize(64);
```

ACV and ACI channels are independently configurable.

#### 4.6.3 Resistance Measurement

The following example adds a DMM channel to the session for two wire resistance measurement and sets its range:

```
//Add a 2-wire resistance channel in the 1 kohm range.
dmmSession.CreatedMMChannel("pdna://192.168.100.2/Dev0/Ai0", 1000,
    UeiDMMModeTwoWiresResistance);
```

Resistance modes:

- `UeiDMMModeTwoWiresResistance` – Measure 2-wire resistance
- `UeiDMMModeFourWiresResistance` – Measure 4-wire resistance

See **Table 2-5** for a list of available resistance ranges. If you pass in an unsupported range value, the board defaults to the next available range. For example, if you request a range of 20 kΩ, the board will instead use the 100 kΩ range.



**4.6.4 Auto-Ranging** Auto-ranging is useful for measuring signals with an unknown or varying range. In auto-ranging mode, the DMM automatically switches the range up if the input is overrange or down if the input is under-range. Auto-ranging is supported when using the voltage, current, or resistance measurement modes. When using the UeiDaq Framework API, auto-ranging is enabled by default.

Auto-ranging can also be enabled or disabled using the following method (enabling shown).

```
//Enable auto-ranging
acv->EnableAutoRange(TRUE);
```

**4.6.5 Guardian Diagnostic Measurement** The following example configures the session for diagnostics. The range input parameter is ignored for Guardian measurements.

```
//Add DMM channel for temperature diagnostics.
dmmSession.CreatedMMChannel("pdna://192.168.100.2/Dev0/Ai0", NULL,
    UeiDMMModeGuardianTemperature);
```

Guardian channels:

- UeiDMMModeGuardianDCCurrent – Guardian DC current (amps)
- UeiDMMModeGuardianACCurrent – Guardian AC current (Arms)
- UeiDMMModeGuardianVoltage0 – -17.5 Volt supply
- UeiDMMModeGuardianVoltage1 – +17.5 Volt supply
- UeiDMMModeGuardianVoltage2 – -5.5 Volt supply
- UeiDMMModeGuardianVoltage3 – +5.5 Volt supply
- UeiDMMModeGuardianTemperature – Temp. in the ADC IC (°C)

**NOTE:** When using Framework, we recommend performing Guardian measurements separately from normal V, I, and R measurements. This is because the Guardian configuration may overwrite the measurement channel configuration if they are in the same channel list.

**4.6.6 Add Time Stamp** You can time stamp the data by adding a `ts` channel as the last channel in the resource string: `"pdna://192.168.100.2/Dev0/Ai0,ts"`. The time stamp units will be in seconds. Note that there are no spaces in a properly formatted resource string.



**4.6.7 Configure Input Disconnection** Inputs are disconnected if a read has not been performed within a set amount of time (default is 10 seconds). The next read command reconnects the input. Relays will have some settling time when they are switched back on; this can be avoided by setting the timeout period greater than the delay between reads. The following code configures the timeout period for the entire layer

```
//Set input disconnection timeout to 20 seconds.
dcm->SetDisconnectTimeout(20);
```

The minimum timeout is 1 second and the maximum is 250 seconds.

**NOTE:** The DNx-DMM-261 automatically disconnects and reconfigures inputs in the event of DC over-voltage. Over-voltage protection is always enabled and cannot be configured by the user.

**4.7 Configure the Timing** Only Point-by-Point data acquisition mode can be used to transfer data between a Framework application and the DNx-DMM-261.

Point-by-Point mode transfers one sample at a time to/from each configured channel of the I/O board. The delay between samples is controlled by the host application (e.g., by using a Sleep function), thus limiting the data transfer rate to a maximum of 100 Hz. This mode is also known as immediate mode or simple mode.

```
//configure session to use Point-by-Point DAQ mode
dmmSession.ConfigureTimingForSimpleIO();
```

**4.8 Start the Session** After the session is configured, it can be started manually:

```
//Start the session.
dmmSession.Start();
```

If the session isn't explicitly started, it will start automatically the first time that data is transferred.

**4.9 Read Data** The `CUEiDMMReader` class provides the methods listed in **Table 4-2** for reading data from the DMM-261. Additional methods that provide information about the status of the input protection system are described in Section 4.9.1.

**Table 4-2 CUEiDMMReader Class Methods**

Method	Description
ReadSingleScan	Reads a single scan in the configured measurement mode and stores data in a supplied buffer
ReadMultipleScans	Reads a specified number of scans in the configured measurement mode and stores data in a supplied buffer.



**Table 4-2 CUeiDMMReader Class Methods**

ReadSingleScanAsync	Reads a single scan asynchronously in the configured measurement mode and stores data in a supplied buffer
ReadMultipleScansAsync	Reads a specified number of scans asynchronously in the configured measurement mode and stores data in a supplied buffer
ReadStatus	Gets the status word associated with the last data read. Status bits are as follows: <ul style="list-style-type: none"> <li>- 0001 - protection tripped in max range</li> <li>- 0010 - protection - DMM will retry</li> <li>- 0100 - protection tripped but DMM reconfigured for safe range</li> <li>- 1000 - protection tripped in max range</li> </ul>
ReadSingleScanExt	Reads scaled data with extended status information

The following example code reads scaled data from all configured channels:

```

//Create a reader and link it to the session's data stream.
CUeiDMMReader reader(dmmSession.GetDataStream());

//Buffer must be large enough to contain one sample per channel.
double* data= new double [dmmSession.GetNumberOfChannels()];

//Read one sample per channel.

reader.ReadSingleScan(data);
    
```

**4.9.1 Read Input Protection Status**

You can read the status of the input protection system, which automatically disconnects DCV, ACV, and Resistance inputs in case of over-voltage. Status is only updated when channel data is read, e.g., when `ReadSingleScan()` is called.

The status bits can be checked using the following `CUeiDMMReader` methods:

- `IsProtectionTripped()` – Returns `true` if protection has tripped. The inputs will be disconnected. The firmware will reconfigure the range and reconnect unless it is already in the maximum range. This bit is cleared when the DMM-261 returns to the user range.
- `IsProtectionTrippedMaxRange()` – Returns `true` if protection has tripped in the max range. Inputs are disconnected until the session is restarted.
- `IsProtectionReconfiguredSafeRange()` – Returns `true` if protection has tripped but the DMM-261 has reconfigured to a safe range to continue collecting data. The DMM-261 will attempt to re-enter the user-configured range once the input signal is within a safe limit.



```
//Print the input protection status returned with the last read data.

if (reader.IsProtectionTrippedMaxRange())
    std::cout <<
        "Protection tripped in max range.";
else if (reader.IsProtectionReconfiguredSafeRange())
    std::cout <<
        "Protection tripped and reconfigured DMM-261 to a safe range.";
else if (reader.IsProtectionTripped())
    std::cout <<
        "Protection tripped. Will reconfigure to a safe range if possible.";
```

#### 4.10 Stop the Session

The session will automatically stop and clean itself up when the session object goes out of scope or when it is destroyed. To manually stop the session:

```
//Stop the session.

dmmSession.Stop();
```

To reuse the object with a different set of channels or parameters, you can manually clean up the session as follows:

```
//clean up session and free resources

dmmSession.CleanUp();
```

#### 4.11 Using the DMM-261 with the MUX-461

If your DMM-261 is connected to one or more DNx-MUX-461 Series boards, your application can use the `SetMuxDeviceCount()` method to create a `MuxGroup`. A `MuxGroup` may consist of up to five DNx-MUX-461 Series boards that can provide input to a single DMM-261 board. The `MuxGroup` API ensures that within a `MuxGroup`, only one MUX channel at a time is connected to a DMM-261, thereby safely connecting the MUXes to the DMM so that damage to the equipment is prevented.

In order to create a `MuxGroup` in a system with a DNx-DMM-261, a `CUeiDMMChannel` object must be created using a separate session and then calling `SetMuxDeviceCount()`.

```
// Use a CUeiDMMChannel object to create a MuxGroup with 5 MUXes

CUeiDMMChannel* dmmChannel = dmmSession.CreatedMMChannel
    ("pdna://192.168.100.2/Dev0/ai0",
     10,
     UeiDMMModeCalibratedDCVoltage);

dmmChannel->SetMuxDeviceCount(5);
```



Use `WriteDMMuxConfig()` to connect a DNx-MUX-461 Series Multiplexer Board to the DNx-DMM-261. DNx-MUX-461 Series Multiplexer Boards have two relay banks (A and B), each with up to 13 channels, thereby providing a total of up to 26 channels. Individual MUX input channels are selected by specifying a channel number and relay bank..

```
CUeiDMMWriter::WriteDMMuxConfig
    (muxDeviceNum, muxChannel, muxRelaySelect.

// muxDeviceNum      index of the MUX in the system (1-5)
// muxChannel        channel number (0-12 for MUX-461,
//                  0-11 for MUX-461-350)
// muxRelaySelect    tUeiMuxRelaySelect
```

A separate session to connect to a DNx-MUX-461 Series board is not needed unless you want to directly configure or read status from the board. See the *“DNx-MUX-461 Series User Manual”* for more information.

**NOTE:** Using the UeiDaq Framework MuxGroup API is best suited for applications where only one type of DMM measurement is needed since the DMM relays on the DNx-MUX-461 boards are set when the MuxGroup is created. If multiple signal types to read, the low-level API should be used to create MuxGroups.



## Chapter 5 Programming with the Low-level API

This chapter provides the following information about programming the DNx-DMM-261 using the low-level API:

- About the Low-level API (Section 5.1)
- Example Code (Section 5.2)
- Data Acquisition Modes (Section 5.3)
- Point-by-Point API (Section 5.4)

### 5.1 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 the Windows OS, we recommend that you use the UeiDaq High-level Framework API (see Chapter 4). The Framework simplifies the low-level API, making programming easier and faster while still providing access to the majority of Low-level API features. 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: *<PowerDNA-x.y.z>/docs*
- On Windows: *C:\Program Files (x86)\UEI\PowerDNA\Documentation*

**NOTE:** The DNx-DMM-261 is supported in PowerDNA version 5.2+. If you're unsure if your version supports the board, please contact Technical Support at [support@ueidaq.com](mailto:support@ueidaq.com)

### 5.2 Example Code

Application developers are encouraged to explore the self-documented source code examples to get started programming UEI products. The example code is located in the following directories:

- On Linux: *<PowerDNA-x.y.z>/src/DAQLib\_Samples*
- On Windows: *C:\Program Files (x86)\UEI\PowerDNA\SDK\Examples*

The I/O board number is embedded in the name of the example code. For example, the Sample261 folder contains example code specific to the DNx-DMM-261. The example code should run out of the box after inputting the IOM's IP address and the board's Device Number (DEVN).



### 5.3 Data Acquisition Modes

The following data acquisition (DAQ) mode is available for transferring data between the DNx-DMM-261 and the low-level user application:

- **Point-by-Point:** Transfers one data point at a time to/from each configured channel of a single I/O board. Timing is controlled by the user application, which limits the transfer rate to 100 Hz. This mode is also known as immediate mode or simple mode.

Please refer to “FAQ - Data Acquisition Modes” for an overview and comparison of all the different acquisition modes offered by UEI. The “PowerDNx Protocol Manual” includes more detailed information about the protocols. Both of these documents are located in the directories listed in Section 5.1.

### 5.4 Point-by-Point API

Low-level functions specific to the DNx-DMM-261 are summarized in **Table 5-1**. If your DMM-261 is connected to one or more DNx-MUX-461 Series boards as described in Section 2.11, your application can also use the MuxGroup API functions listed in **Table 5-2**. See *Sample261* and *Sample261MuxGroup* for comprehensive examples which include typical initialization, error handling, and usage of these functions.

Note that some low-level API functions take a channel list as an argument. Each entry in the channel list specifies a DMM measurement channel and a range for that channel. The channels in the list can be from the same DMM mode, e.g., `DQ_DMM261_VDC_CAL_CH` and `DQ_DMM261_VDC_ACC_DIV_CH`. However, you cannot mix and match channels from different DMM modes (e.g., `DQ_DMM261_VDC_CAL_CH` and `DQ_DMM261_VAC_RMS_CH`). If you want to take a different type of measurement, call `DqAdv261Read()` or `DqAdv261MuxGroupSet()` again with a new channel list.

If the channel list specifies the channels for RMS AC Current and Averaged RMS AC Current or RMS AC Voltage and Averaged RMS AC Voltage, the API will only return the averaged data for both.

Low-level functions and their parameters as well as available DMM measurement channels are described in detail in the “*PowerDNA API Reference Manual*”.



**Table 5-1 Low-level API for the DNx-DMM-261**

Function	Description
DqAdv261Read	This function can read all available measurement and diagnostic data from the DMM-261.
DqAdv261ReadStatus	Reserved for UEI use.
DqAdv261AutoZero	Auto-zero DC voltage and DC current channels.
DqAdv261SetResRange	Set the resistance measurement mode and range.
DqAdv261SetConfig	Configure averaging, NPLC, AC zero crossing detection, and input protection.
DqAdv261SetFixedZeroCrossingLevel	Configures a user-defined fixed zero-crossing level for AC voltage or current measurements. <b>Note:</b> This selection is not compatible with auto-ranging since the range needs to be passed in as a parameter

**Table 5-2 MuxGroup API Functions**

Function	Description
DqAdv261MuxGroupCreate	Link a DMM-261 and up to five MUX layers into a group.
DqAdv261MuxGroupDestroy	Release DMM and all MUX layers from a MuxGroup.
DqAdv261MuxGroupDisconnect	Disconnect all MUX channels from the DMM-261 without connecting a new input channel or destroying the MuxGroup.
DqAdv261MuxGroupSet	Safely configure the MuxGroup relays for the specified MUX input channel and DMM mode before a call to DqAdv261MuxGroupRead().
DqAdv261MuxGroupRead	Read data from a DMM that is part of a MuxGroup.



# Appendix A

## Accessories

### A.1 Cables

#### DNA-CBL-261-BNC

DNA-CBL-261-BNC cables are 12 inches long and provide connection to field devices using standard BNC connectors.

#### DNA-CBL-261

DNA-CBL-261 cables are 10 inches long and provide connection to field devices using standard banana-style connectors.

#### DNA-CBL-37

The DNA-CBL-37 is a 37-conductor flat ribbon cable with 37-pin male D-sub connectors on both ends. The cable is 3 ft long (0.91 m) and weighs 3.4 oz. (98 g).

#### DNA-CBL-37S-x

The DNA-CBL-37S-x cables are 37-way round shielded cables with male to female connectors. DNA-CBL-37S-x cables are available in 1 ft, 3 ft (DNA-CBL-37S), 5 ft, 10 ft, and 20 ft lengths.

#### DNR-CBL-461-x (x can be 1, 2, 3, 4, or 5)

The DNR-CBL-461-x cable connects a DNR-DMM-261 board to as many as five DNR-MUX-461 Series boards inside of a Rack chassis. The cable plugs into the JPOW1 connectors on the PCBs in a daisy-chain configuration as described in Section 2.11.2. Five different cable options are available, where x corresponds to the number of DNR-MUX-461 Series boards in the system.

### A.2 Screw Terminal Panels

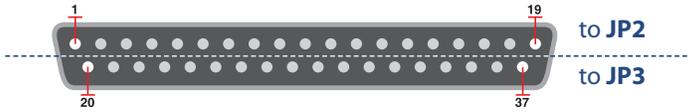
#### DNA-STP-37

The DNA-STP-37 provides easy screw terminal connections for all DNA and DNR series I/O boards which utilize the 37-pin connector scheme. The DNA-STP-37 is connected to the I/O board via either DNA-CBL-37 or DNA-CBL-37S series cables. The dimensions of the STP-37 board are 4.2w x 2.8d x 1.0h in. (10.7 x 7.1 x 2.5 cm) (with standoffs). The weight of the STP-37 board is 2.4 ounces (69 grams).



**JP1 — DB-37 (male)  
 37-pin connector:**

to JP3	20	1	to JP2
to JP3	21	2	to JP2
to JP3	22	3	to JP2
to JP3	23	4	to JP2
to JP3	24	5	to JP2
to JP3	25	6	to JP2
to JP3	26	7	to JP2
to JP3	27	8	to JP2
to JP3	28	9	to JP2
to JP3	29	10	to JP2
to JP3	30	11	to JP2
to JP3	31	12	to JP2
to JP3	32	13	to JP2
to JP3	33	14	to JP2
to JP3	34	15	to JP2
to JP3	35	16	to JP2
to JP3	36	17	to JP2
to JP3	37	18	to JP2
	19		to JP2

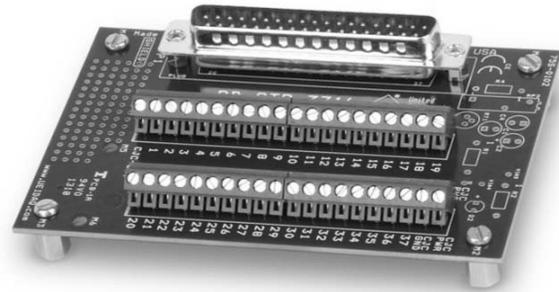


**JP2 — 20-position  
 terminal block:**

19	19
18	18
17	17
16	16
15	15
14	14
13	13
12	12
11	11
10	10
9	9
8	8
7	7
6	6
5	5
4	4
3	3
2	2
1	1
CABLE SHIELD	

**JP3 — 20-position  
 terminal block:**

N/C	N/C
37	37
36	36
35	35
34	34
33	33
32	32
31	31
30	30
29	29
28	28
27	27
26	26
25	25
24	24
23	23
22	22
21	21
20	20



**Figure A-1 Pinout and Photo of DNA-STP-37 Screw Terminal Panel**

