United Electronic Industries

The High-Performance Alternative

DNx-AO-318
DNx-AO-318-020
DNx-AO-318-024
User Manual

AO-318 8-channel, 16-bit, isolated D/A voltage board with built-in-test,
AO-318-020 8-channel, 16-bit, isolated D/A current board with built-in-test and
AO-318-024 8-channel, 16 bit, isolated D/A current board with built-in-test
for the PowerDNA Cube and RACK series chassis

January 2020

PN Man-DNx-AO-318

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

Contacting United Electronic Industries

Mailing Address:

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

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

Support:

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

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

Internet Support:

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

Product Disclaimer:

WARNING!

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

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

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

Table of Contents

List of Figures

Chapter 1 Introduction

This manual outlines the feature set and use of the DNx-AO-318, an 8-channel digital-to-analog output board:

- AO-318-1: 16-bit, 8-channel, ±10 V analog voltage output board
- AO-318-020: 16-bit, 8-channel, 0-20 mA analog current output board
- AO-318-024: 16-bit, 8-channel, 0-24 mA analog current output board

The following sections are provided in this chapter:

- Organization of this Manual (Section 1.1)
- AO-318 Board Overview (Section 1.2)
- Features (Section 1.3)
- Specification (Section 1.4)
- Device Architecture (Section 1.5)
- Indicators (Section 1.6)
- Wiring & Connections (pinout) (Section 1.7)
- PowerDNA Explorer for AO-318 (Section 1.8)

1.1 Organization of this Manual

This AO-318 User Manual is organized as follows:

- **Introduction**
  Chapter 1 provides an overview of DNx-AO-318 features, device architecture, connectivity, and logic.
- **Programming with the High-Level API**
  Chapter 2 provides an overview of the how to create a session, configure the session, and interpret results with the Framework API.
- **Programming with the Low-Level API**
  Chapter 3 is an overview of low-level API commands for configuring and using the AO-318 series board.
- **Appendix A - Accessories**
  The appendix provides a list of accessories available for use with the DNx-AO-318 board.
- **Index**
  This is an alphabetical listing of the topics covered in this manual.

NOTE: A glossary of terms used with the PowerDNA Cube/RACK and I/O boards can be viewed or downloaded from www.ueidaq.com.

Manual Conventions

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

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

NOTE: Notes alert you to important information.

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

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

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

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

Examples of Manual Conventions

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.

Usage of Terms

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

1.2 AO-318 Board Overview

The DNx-AO-318 boards are fully isolated, high-precision, 8-channel analog output boards.

DNA-AO-318, DNR-AO-318, and DNF-AO-318 board versions are compatible with the UEI Cube, RACKtangle, and FLATRACK chassis respectively. These board versions are electronically identical and differ only in mounting hardware. The DNA version is designed to stack in a Cube chassis. The DNR/F versions are designed to plug into the backplane of a RACK chassis.

1.2.1 DAC Output Capabilities

The boards offer full 16-bit resolution and guarantee monotonicity over the entire operating temperature range. AO-318 boards provide output as follows:

- For the **AO-318** model each channel provides an output range of ±10 V and is capable of driving up to ±10 mA (max).
- For the **AO-318-020** each channel provides an output range of 0-20 mA (sourcing) and is capable of driving up to 470 Ω load.
- For the **AO-318-024** each channel provides an output range of 0-24mA (sourcing) and is capable of driving up to a 400 Ω load.

For applications requiring higher output current or voltage, please refer to the DNx-AO-308-35x series boards.

1.2.2 Guardian Diagnostic Support

The DNx-AO-318 provides extensive built-in-test diagnostics.

An onboard A/D converter on each channel allows the user to monitor diagnostic voltages, current, and temperature. A solid state relay on each output allows the D/A channel to be disconnected from the field I/O so that a complete board self-test can be completed without driving the circuitry connected to the outputs. This relay, in combination with the output current and voltage sensing, can also be set to disconnect the D/A output in the event of an external fault condition such as a short to ground or a DC power supply.

1.2.3 Data Update Rates

A 1024 sample FIFO allows each D/A to be updated at 10 kHz without data loss. Double buffering the outputs combined with the use of low glitch D/As make the DNx-AO-318 an ideal solution for generating low frequency waveforms or providing highly accurate switched stimulus.

1.2.4 Isolation

The outputs of the AO-318 are fully isolated from each other, the I/O chassis as well as from other I/O boards within the I/O chassis. All connections are made through a 37-pin D connector. Users may also connect the DNx-AO-318 boards to our DNA-STP-37 screw terminal panel via the DNA-CBL-37 cable. The cables are fully shielded and are available in 2.5, 10 and 20 foot lengths.

1.2.5 Software Support

Software included with the DNx-AO-318 provides a comprehensive yet easy to use API that supports Windows, Linux, QNX, VxWorks, and most other popular real-time operating systems. Windows users may take advantage of the UEIDAQ Framework, which provides a simple and complete software interface to all popular Windows programming language and data acquisition and control applications, such as LabVIEW and MATLAB.

1.3 Features

The AO-318 boards have the following features:

- 8 independent fully isolated 16-bit digital-to-analog converters
- AO-318-1 design includes dual, redundant DACs per channel (one active at a time); AO-318-020 is designed with a single DAC per channel
- Built-in-test functionality monitors both output voltage and current
- Electronic circuit breakers on under/over-voltage/current/temperature
- 10 kHz per channel max update rate
- AO-318 output range of ±10 V, up to ±10 mA (max) per channel
- AO-318-020 output range of 0-20 mA per channel, up to 470 Ω load
- AO-318-024 output range of 0-24 mA per channel, up to 400 Ω load
- Glitch-free output
- Tested to withstand 5g Vibration, 50g Shock, -40 to +85°C Temperature, and Altitude up to 70000 ft or 21000 meters.

1.4 Specification

The technical specification for the DNx-AO-318 boards are listed below.

Table 1-1. DNx-AO-318 Technical Specifications

Analog Outputs	8 channels
Resolution	16-bits
Max Update Rate:	10 kHz/channel (80 kHz max aggregate)
FIFO Buffer Size	1024 samples
INL (no load)	±6 LSB (0.018%)
DNL (no load)	±2 LSB (0.006%)
Monotonicity	16 bits guaranteed over temperature
Gain Calibration Error	±610 µV, typ.
Offset Calibration Error	±305 µV, typ.
Output Range	±10 V
Output Impedance	0.5 Ω (typ)
Current Drive	±10 mA/channel
Settling Time	50 µs to 16 bits
Slew Rate	1 V/µs
Power up state	0 V ±10 mV
Output Monitoring	
Accuracy (V / I)	±5 mV / 100 µA (± 50 mV / 1 mA over FS temp)
Sample/Update rate	~1 sample/sec on each channel (default)
Isolation	350Vrms channel-to-channel and field wiring to chassis.
Power Consumption	4.0 W (not including output loads)
Operating Temp. (tested)	-40 °C to +85 °C
Operating Humidity	95%, non-condensing

Table 1-2. DNx-AO-318-020 Technical Specifications

Number of Channels	8
Resolution	16 bits
Max Update Rate:	10 kHz/channel (80 kHz max aggregate)
Buffer Size	1K samples (each channel)
INL (no load)	±6 LSB (0.018%), typical
DNL (no load)	±2 LSB (0.006%), typical
Monotonicity Over Temp	16 bits guaranteed
Gain Linearity Error	0.002%
Gain Calibration Error	±2.5 µA, typ.
Offset Calibration Error	±2.5 µA, typ.
Offset Drift	10 ppm/°C, typical
Gain Drift	10 ppm/°C, typical
Output Range	0-20mA
Settling Time	500 µs to 16 bits
Load range	0 to 470 Ohms for full 0-20 mA swing
Isolation	350Vrms
Built-in Test	
Voltage accuracy	+/- 25 mV
Current accuracy	25 µA
Sample rate	Up to 20 Hz per channel total
Power Consumption	4.5 Watt (not including output load)
Operating Temp. (tested)	-40°C to +85°C
Operating Humidity	0 - 95%, non-condensing

Table 1-3. DNx-AO-318-024 Technical Specifications

Number of Channels	8
Resolution	16 bits
Max Update Rate:	10 kHz/channel (80 kHz max aggregate)
Buffer Size	1K samples (each channel)
INL (no load)	±6 LSB (0.018%), typical
DNL (no load)	±2 LSB (0.006%), typical
Monotonicity Over Temp	16 bits guaranteed
Gain Linearity Error	0.002%
Gain Calibration Error	±5 µA typical,
Offset Calibration Error	±5 µA typical,
Offset Drift	15 ppm/ºC, typical
Gain Drift	15 ppm/ºC, typical
Output Range	0-24 mA
Settling Time	500 µs to 16 bits
Load range	0 to 400 Ohms for full 0-24 mA swing
Isolation	350 Vrms
Built-in Test	
Voltage accuracy	+/- 25 mV
Current accuracy	25 µA
Sample rate	Up to 6 Hz per channel total (3 Hz if both current and voltage are monitored)
Power Consumption	4.5 Watt not including output load
Operating Temp. (tested)	-40ºC to +70ºC for full output to 24 mA -40ºC to +85ºC for 0 - 20 mA range
Operating Humidity	0 - 95%, non-condensing
Vibration *IEC 60068-2-6* *IEC 60068-2-64*	5 g, 10-500 Hz, sinusoidal 5 g (rms), 10-500Hz, 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
Altitude	120,000 ft
MTBF	480,000 hours

1.5 Device Architecture

Figure 1-1 is a block diagram of the architecture of the AO-318 board.

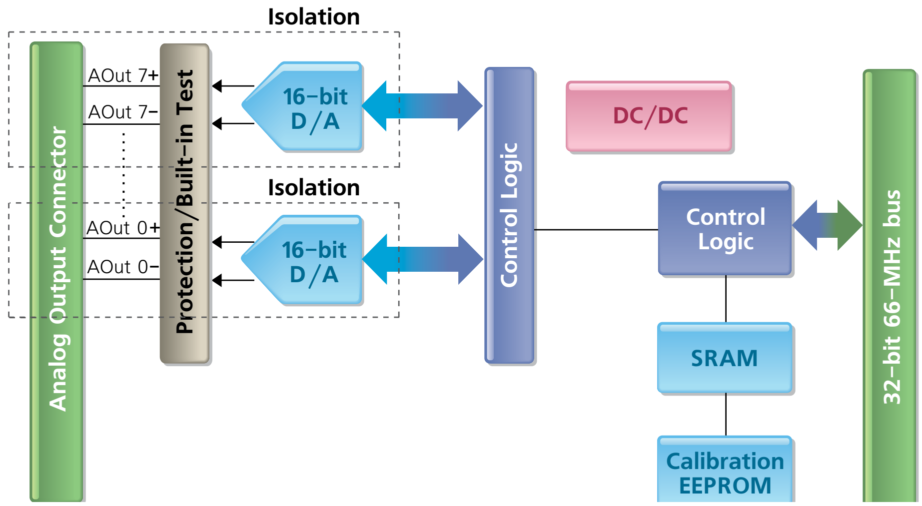

Figure 1-1 Block Diagram of the AO-318

1.5.1 AO-318 Output Circuitry

The AO-318 has eight individual analog output channels.

Each output channel includes a dedicated 16-bit D/A converter and a dedicated A/D converter to monitor voltage, current, and/or temperature for Guardian diagnostic functions (**Figure 1-2** and **Figure 1-3**). Channels are also equipped with dedicated, configurable circuit breakers.

Each output channel is isolated from other channels, and the AO-318 board itself is also optically isolated from the other boards in the chassis.

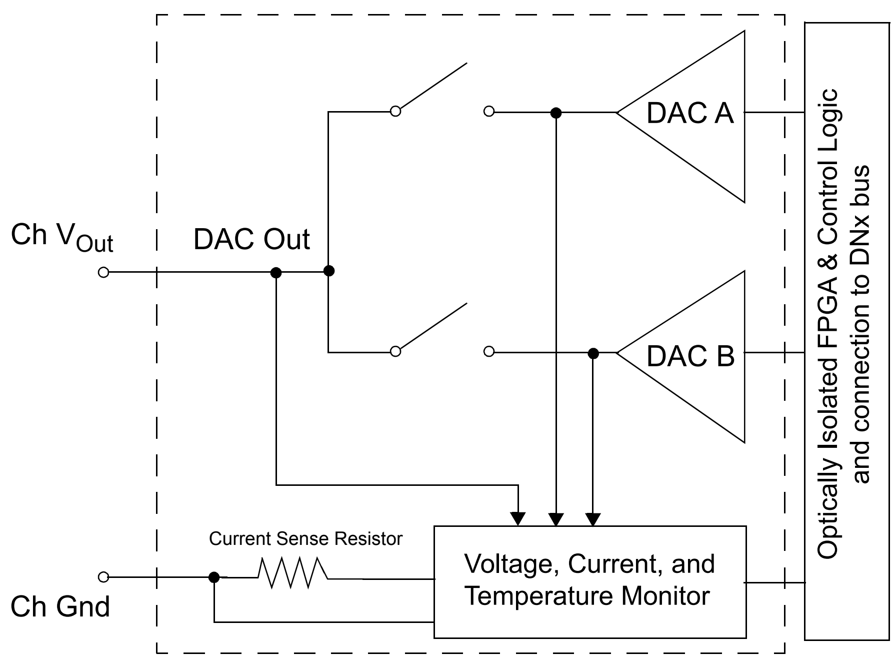

Figure 1-2 Simplified Block Diagram of an AO-318 Channel

On the AO-318, the output DAC is dual-redundant and is software selectable as completely disabled, A, or B.

1.5.2 AO-318-02x Output Circuitry

On the AO-318-02x the output DAC is always set by firmware control to DACA (DACB is not included).

Note that the AO-318-02x current readings for Guardian monitoring are valid within a few microamps. The voltage reading may read lower because of the voltage drop across the DNA-CBL-37 or similar cable which is not accounted for in the voltage readings.

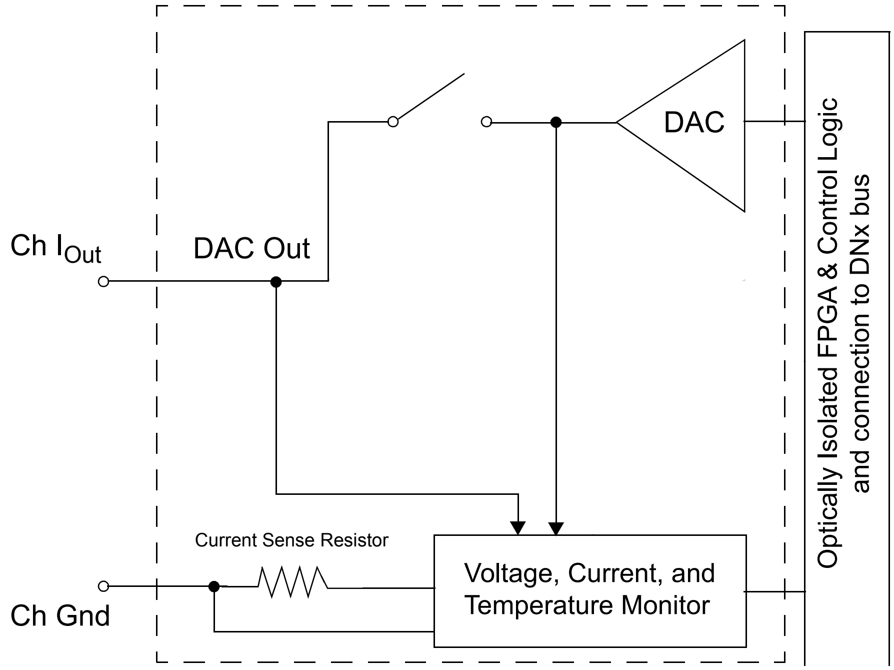

Figure 1-3 Simplified Block Diagram of an AO-318-02x Channel

1.5.3 Guardian Diagnostic Measurements

The Guardian diagnostic readings are performed in the "Voltage, Current, and Temperature Monitor" block (shown in **Figure 1-2** and **Figure 1-3**).

Each of the eight AO-318 output channels is equipped with its own Guardian ADC. The following diagnostic measurements can be read as analog input values on the diagnostic ADC channels:

ADC Diagnostic Channel	AO-318 Diagnostic Measurement	AO-318-020 Diagnostic Measurement
0	Shunt between output connector and A_{GND}	Shunt between output connector and A_{GND}
1	DAC_A output before relay	DAC buffer output
2	DAC_B output before relay	DAC output before the relay
3	DAC output after the relay at output connector	DAC output after the relay at output connector
4	On-die temperature sensor	On-die temperature sensor

Table 1-4. AO-318 Diagnostic Channels for Guardian Diagnostic Measurements

1.5.3.1 Configuring Guardian Diagnostic Readings

The Guardian ADC hardware consists of a 16-bit precision, sequential ΔΣ analog-to-digital converter, which is controlled by the on-board logic.

The sample rate of the Guardian ADC can be set to 5, 7.5 (default), 10, 20, 30, 40, 50 Hz. Note that the ADC uses sequential (multiplexed) inputs. Reading all diagnostic inputs using the default sample rate results in ~1.25 Hz/channel.

UEI recommends using a low sampling rate (the default) when measuring diagnostic values to allow the sequential ADC sufficient time to propagate through all measurements. Higher sampling frequencies yield higher errors, as shown in **Figure 1-3**.

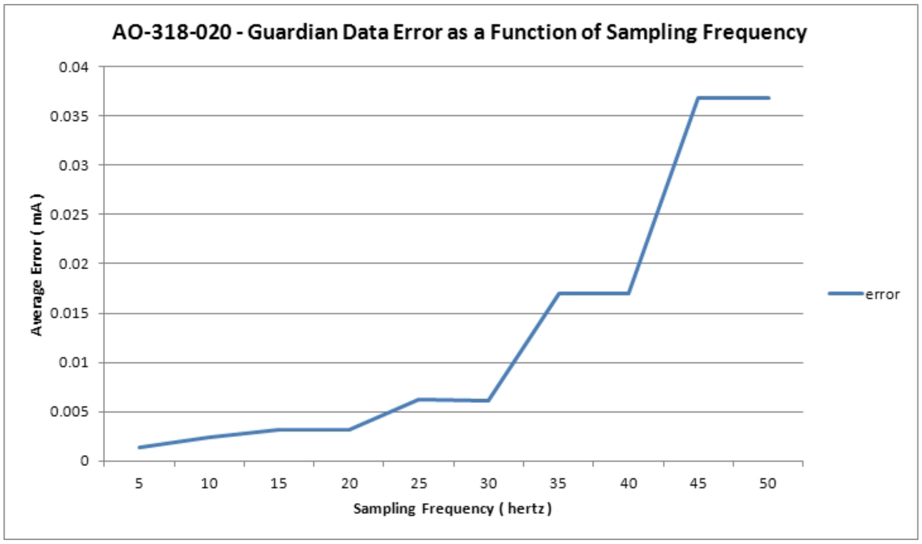

Figure 1-4 Guardian ADC Error Rate vs Sampling Frequency

1.5.4 Circuit Breakers

Relay(s) at the output of the digital-to-analog converters act as circuit breakers. Refer to **Figure 1-2** and **Figure 1-3** for location.

Relays are controlled by the on-board logic, which opens and/or closes relays when the user-programmable minimum / maximum conditions are exceeded.

Each output channel on the AO-318 standard model has 2 circuit breakers due to dual redundant DACs, and each output on the AO-318-020 has one.

Each circuit breaker can be controlled by up to 2 sources that are user-programmable. Users can set a minimum or maximum value for any of the 5 diagnostic ADC channel measurements listed in **Table 1-4** to use as a control condition (or the min/max rule can be disabled).

ADC Diagnostic Channel		Circuit Breaker Min/Max for ADC CHs		Circuit Breaker Control Source Selection
ADC0	→	CB Min/Max for ADC0	←	$DAC_{A\|B}$ CB Selector 0
ADC1	→	CB Min/Max for ADC1	←	$DAC_{A\|B}$ CB Selector 1
ADC2	→	CB Min/Max for ADC2	←	020 only CBSelector 2
ADC3				
ADC4				

If a circuit breaker trips, you can disable the circuit breaker channel and/or issue a reset command to the circuit breaker once the signal returns within the min/max bounds.

1.6 Indicators

The DNx-AO-318 and DNx-AO-318-02x indicators are described in **Table 1-5** and illustrated in **Figure 1-5**.

Table 1-5 AO-318 Indicators

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

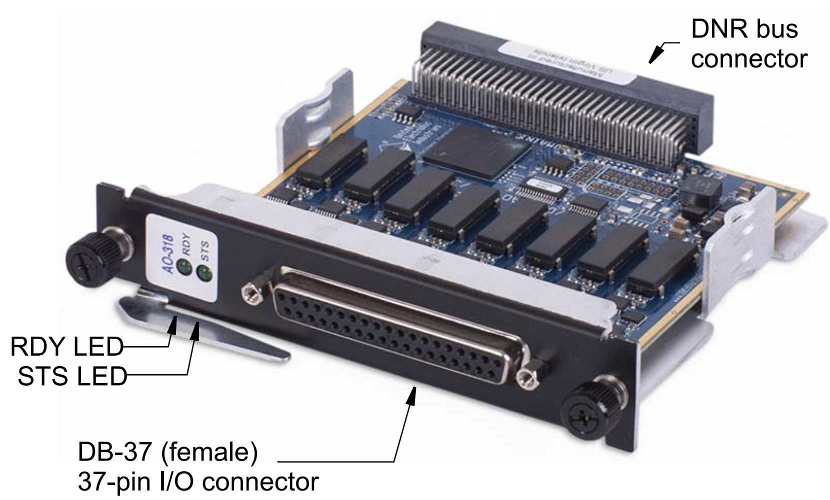

Figure 1-5 Photo of DNR-AO-318 Analog Output Board

1.7 Wiring & Connections (pinout)

Figure 1-6 below illustrates the pinout of the AO-318 and AO-318-02x. Note that analog output signals and analog output grounds on the AO-318 are pin-compatible with AO-308-series output pins.

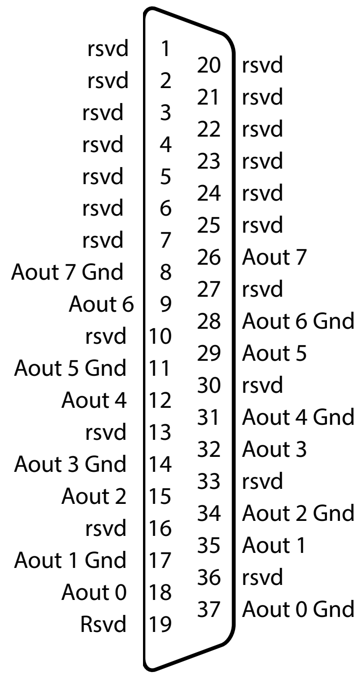

Figure 1-6 Pinout diagram of AO-318

NOTE: Do not connect to any pins marked as reserved (rsvd).

1.8 PowerDNA Explorer for AO-318

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

When using PowerDNA Explorer to explore your AO-318, note that the right-hand panel contains five tabs:

- **Input**: displays input values from the Guardian subsystem
- **Circuit Breaker**: provides circuit breaker state and configuration
- **Output**: sets immediate output values
- **Initialization**: initializes output values at power-up
- **Shutdown**: sets output level applied at shutdown

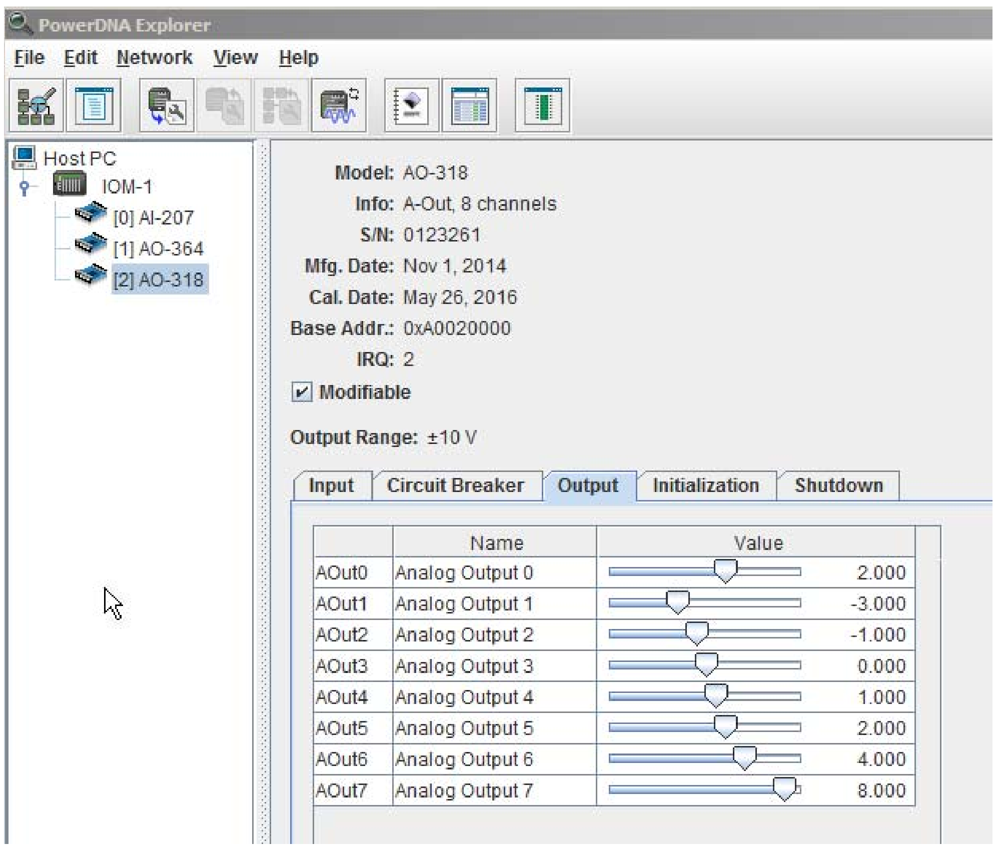

Figure 1-7 PowerDNA Explorer Output Control Panel for AO-318

1.8.1 Guardian Diagnostic (Input) Values

The **Input** tab provides access to Guardian diagnostic readings and contains the following columns:

- **AInx**: read-only display of the channel number.
- **Name**: a name or note that you wish to give to the channel.
- **Guardian Diagnostic Channels, Subchannel 0 through 4**: displays five Guardian ADC Channels as described in Section 1.5.3. The default order is shown below in **Figure 1-8**, but note that this order can be configured with the low-level API using `DqAdv318SetConfig()` function.

Figure 1-8 shows the **Input** tab for the AO-318. Note that the AO-318-02x is a current sourcing board that has a single CB. For the AO-318-02x DAC A and DAC B voltages map to voltages before and after the CB.

In PowerDNA Explorer diagnostic channels display as follows:

ADC Diagnostic Channel	Selector	Input Units
0	Current [mA]	mA
1	DAC A Voltage	Va
2	DAC B Voltage	Vb
3	Output Voltage	Vo
4	Temp. [°C]	°C

To read the Guardian diagnostic values, click the **Read Input Data** button. Note that reading diagnostic input values and circuit breaker status can take from several seconds to several minutes to update.

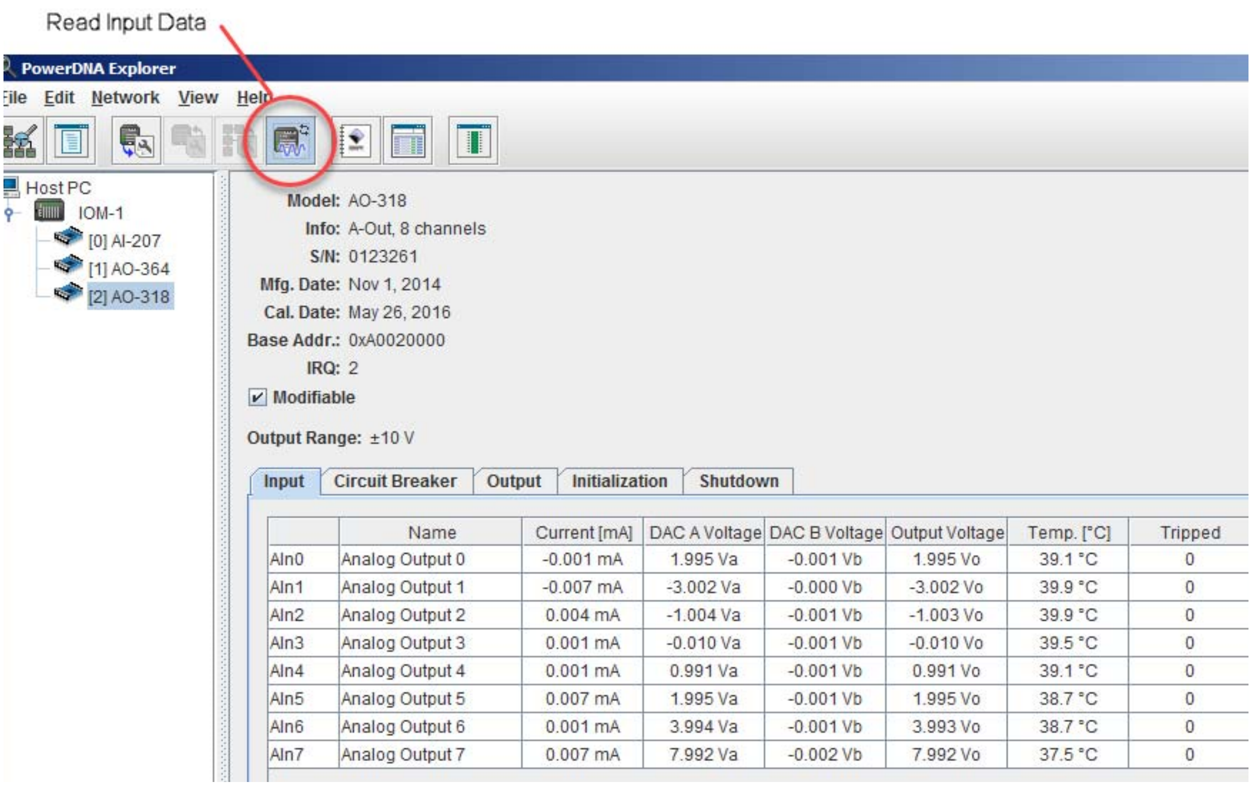

Figure 1-8 PowerDNA Explorer Diagnostic Input Panel for AO-318

1.8.2 Circuit Breaker Values

The **Circuit Breaker** tab allows users to configure channel circuit breakers. The following columns are included:

- **AInx**: read-only display of the channel number.
- **Name**: a name or note that you wish to give to the channel.
- **Tripped**: displays whether the relay is open or closed.
  Note that *Start Reading Input Data* must be running!
- **Reset**: controls whether the relay is open or closed.
- **Min X:** The minimum or underlimit value for Circuit Breaker X.
- **Max X**: The maximum or overlimit value for Circuit Breaker X.
- **Breaker X**: Enable/Disable Circuit Breaker X, and if enabled, choose measurement unit: Current in mA, Output Voltage in V, or Temperature in °C. (See **Figure 1-9**).

When changing the minimum, maximum or other values, you must store the configuration after you enter the changed values. To save the configuration, click the **Store Config** button. To read back the configuration to confirm the settings, click the **Reload Config** button.

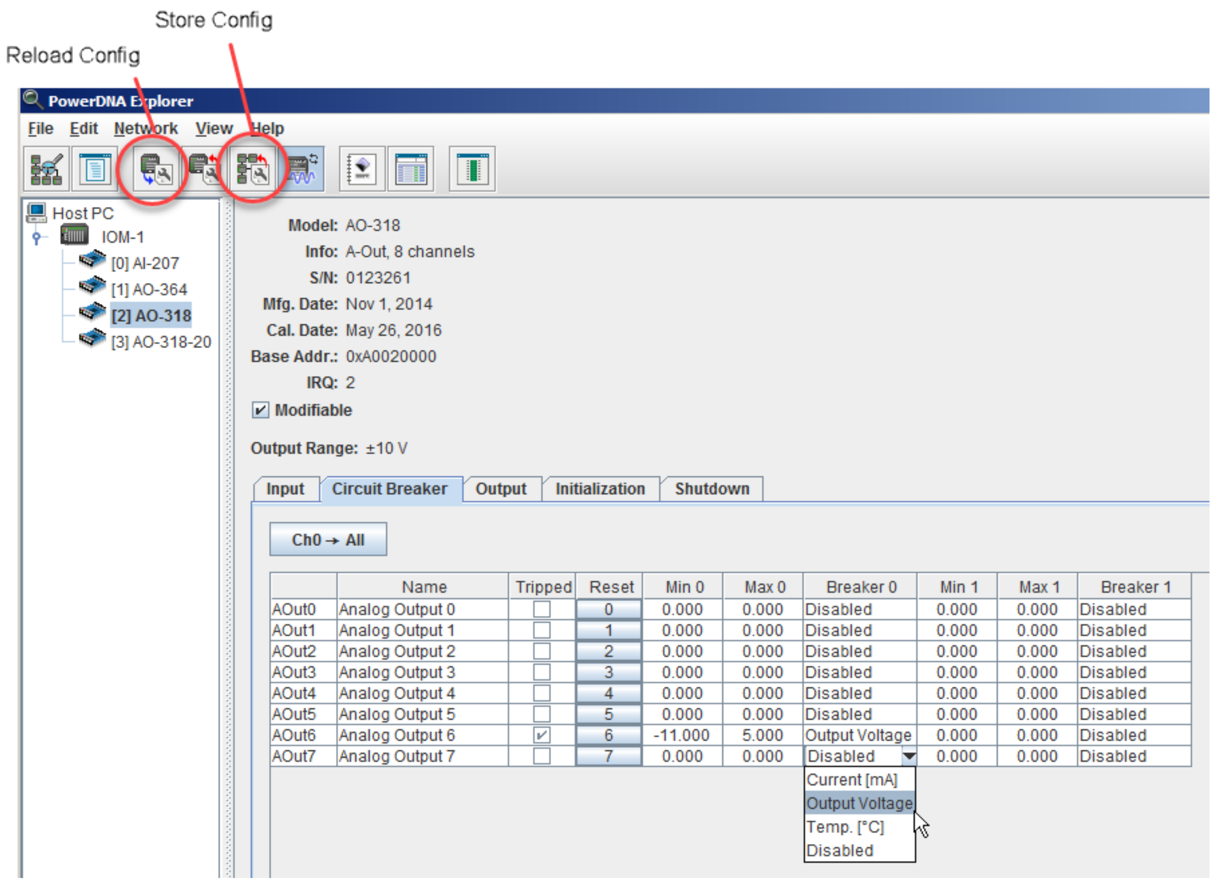

Figure 1-9 PowerDNA Explorer Circuit Breaker Panel for AO-318

1.8.3 Output Values

The **Output** tab, shown **Figure 1-10** below and in **Figure 1-7** on page 12, contains the following columns:

- **AOutx**: read-only display of the channel number.
- **Name**: a name or note that you wish to give to the channel.
- **Value**: slider and numeric text field for immediately setting the voltage (AO-318) or current (AO-318-02x) of the corresponding output channel. The valid value range is shown in the **Output Range** display under the **Modifiable** checkbox.
  This value is written instantaneously when the slider is released or after pressing **Enter** in the numeric field; pressing the **Store Config** button is not required for updating the value in hardware.

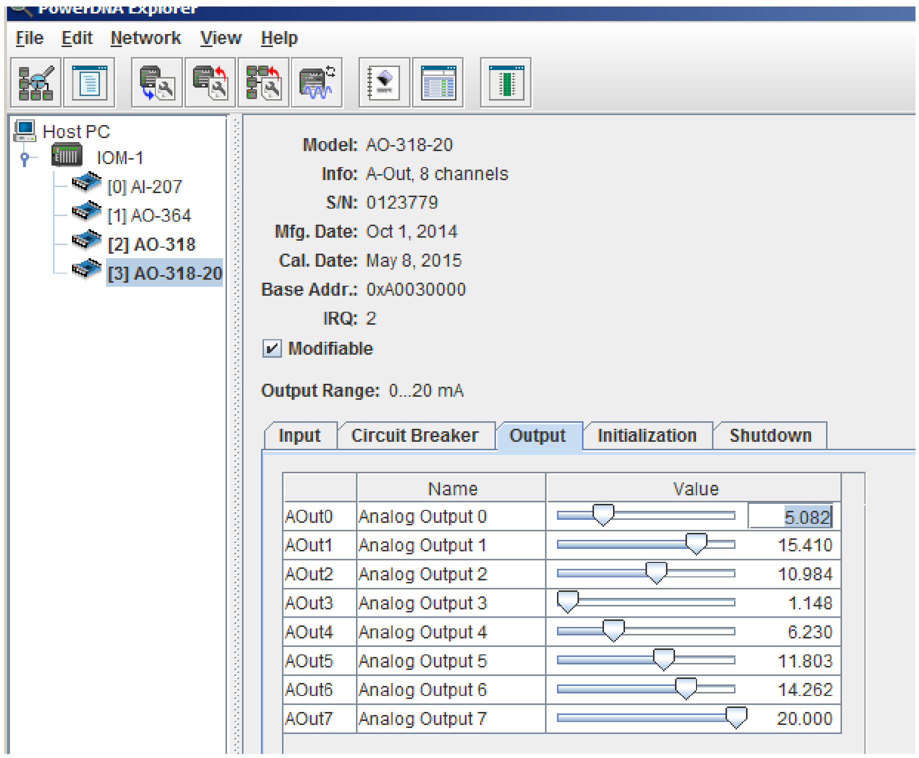

Figure 1-10 PowerDNA Explorer Output Panel for AO-318-020

NOTE: Pressing **Reload Configuration** in v4.9 will read the current output value, which is useful if you just restarted your PowerDNA Explorer. Older or mismatched firmware will use 0 as the default value.

1.8.4 Initialization Values

The **Initialization** tab is similar to the immediate output, shown in **Figure 1-10** and in **Figure 1-7** on page 12, but instead of outputting values immediately, stores values into the EEPROM configuration to set the output value on power-up. The factory default value is 0.

1.8.5 Shutdown Values

The **Shutdown** tab is similar to the immediate output, shown in **Figure 1-10** and in **Figure 1-7** on page 12, but instead of outputting values immediately, stores values into the EEPROM configuration to set the output value in shutdown mode. The factory default value is 0.

Chapter 2 Programming with the High-level API

This chapter provides the following information about using the UeiDaq high-level Framework API to program the DNx-AO-318:

- About the High-level Framework (Section 2.1)
- Creating a Session (Section 2.2)
- Configuring the Resource String (Section 2.3)
- Configuring for Output (Section 2.4)
- Configuring the Timing (Section 2.5)
- Writing Output Data (Section 2.6)
- Monitoring Guardian Diagnostic Measurement (Section 2.7)
- Accessing Circuit Breakers (Section 2.8)
- Cleaning-up the Session (Section 2.9)

2.1 About the High-level Framework

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

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

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

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

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

2.2 Creating a Session

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

```
// create a session object for output
CUeiSession aoSession;
```

2.3 Configuring the Resource String

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

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

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

For example, the following resource string selects analog output lines 0,1,2,3 on device 1 at IP address 192.168.100.2: “pdna://192.168.100.2/Dev1/Ao0:3” as a range, or as a list “pdna://192.168.100.2/Dev1/Ao0,1,2,3”.

2.4 Configuring for Output

The Protected Output session type generates voltage or current out of a Guardian analog output (AO) device and additionally sets up Guardian features.

AO-318 protected output channels provide diagnostic capabilities, such as user-programmable Guardian diagnostic circuit breakers and the ability to monitor diagnostic current, voltage, and temperature measurements.

Note that each analog output channel on the AO-318 is built with redundant DACs (DAC A and DAC B). The AO-318-02x is built with only one DAC (DAC A). Each of these DACs includes a hardware-controlled, user-configured circuit breaker.

NOTE: Refer to Section 2.7 for more information about monitoring Guardian diagnostics and Section 2.8 for more information about accessing circuit breakers.

Use `CreateAOProtectedChannel()` to configure one or more protected analog output channel(s). For example, the following call configures channel 0 of an AO-318 set as device 1 for analog output:

```
// Configure channel 0,1 for output

aoSession.CreateAOProtectedChannel(“pdna://192.168.100.2/Dev1/ao0”,
                                   dacMode,
                                   measurementRate,
                                   enableAutoRetry,
                                   autoRetryRate);
```

`CreateAOProtectedChannel` configures the following parameters:

- **DAC mode**: Selects which DAC is connected to the output channel. Supported values for the AO-318-1 are ‘Disconnected’, ‘DAC A connected’, and ‘DAC B connected’. For the AO-318-02x, supported values are ‘Disconnected’ and ‘DAC A connected’.
- **Measurement rate**: the rate at which the AO device monitors diagnostic channels. This rate has a direct influence on how fast the device reacts to an under or over-current condition. Setting this value too high increases the error rate, as described in Section **Figure 1.5**.
- **Enable auto-retry**: specifies whether the AO device attempts to re-close the circuit, after an over- or under-limit condition was detected to open the circuit, and is now back within an acceptable range.
- **Auto-retry rate**: specifies how often the AO device attempts to close the circuit.

2.5 Configuring the Timing

You can configure the AO-318 to run in simple mode (point by point).

In simple mode, the delay between samples is determined by software on the host computer. In buffered mode, the delay between samples would be determined by the AO-318 on-board clock.

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

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

aoSession.ConfigureTimingForSimpleIO();
```

2.6 Writing Output Data

Writing DC-level data is done using a *writer* object(s).

You can create a writer object that writes raw data straight to the D/A converter. You can also create a writer object that writes data scaled as volts or amps. Framework automatically performs a conversion to binary code before sending the data to the D/A converter.

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

```
// create a writer and link it to the session's stream
CueiAnalogScaledWriter writer(aoSession.GetDataStream());
// the buffer must be big enough to contain one value per channel
double data[2] = {0.0, 0.0};
// write one scan, where the buffer will contain one value per channel
writer.WriteSingleScan(data);
```

Or you can create a writer object that writes raw data straight to the DAC.

```
// create a writer and link it to the session's stream
CueiAnalogRawWriter writer(session.GetDataStream());
// the buffer must be big enough to contain one value per channel
uint16 data[2] = {0x1234, 0x5678};
// write one scan, where the buffer will contain one value per channel
writer.WriteSingleScan(data);
```

All the AO-318 analog output boards are programmed the same way.

2.7 Monitoring Guardian Diagnostic Measurement

Diagnostic measurements are read using an Analog Input session. Up to five analog input channels can be associated with each analog output channel.

- AI channels associated with AO channel 0 are 0, 1, 2, 3, 4
- AI channels associated with AO channel 1 are 8, 9, 10, 11, 12
- and so forth up to AO channel 7 as 56, 57, 58, 59, 60.

Use an Analog Input session the same way you would to measure from any UEI Analog Input device.

You can read all 40 diagnostic measurements with the channel list: “0:4,8:12,16:20,24:28,32:36,40:44,48:52,56:60”.

The following code shows how to read AO channel 0 diagnostic values for an AO-318:

```
// create an input session and configure it for simple timing

CUeiSession aiSession;
aiSession.CreateAIChannel("pdna://192.168.100.2/Dev1/Ai0:4",
                         -10.0, 10.0,
                         UeiAIChannelInputModeDifferential);
aiSession.ConfigureTimingForSimpleIO();

// the buffer must be big enough to contain one value per channel

CUeiAnalogScaledReader aiReader(aiSession.GetDataStream());
double values[5];

// read one scan

aiReader.ReadSingleScan(values);
```

2.8 Accessing Circuit Breakers

Each circuit breaker can monitor up to two of the five diagnostic channels. The circuit breaker has the capability of opening the circuit when one of the diagnostic measurements exceeds the user-configured minimum/maximum values.

This section describes how to configure, get the status of, and manually reset the circuit breakers on the AO-318.

2.8.1 Configuration

You can configure the circuit breaker parameters using the channel object methods (or a property node under LabVIEW). Parameters include which ADC channels, Circuit Breaker channels, Low/High limits, Over/Under limit count.

Identify ADC diagnostic channels: set up to 5 diagnostic channels to read Guardian values. Possible values are:

- `UeiAODiagnosticCurrent`: Monitor output current
- `UeiAODiagnosticDACAVoltage`: Monitor voltage out of DAC A (before relay on AO-318-1)
- `UeiAODiagnosticDACBVoltage`: Monitor voltage out of DAC B (before relay on AO-318-1)
- `UeiAODiagnosticVoltage`: Monitor output voltage (after relay)
- `UeiAODiagnosticTemperature`: Monitor dual DAC die temperature

`UeiAODiagnosticNone`: Do not monitor

```
// Set ADC channel 0 to monitor current; ch is analog output channel

pAOChan =
dynamic_cast<CUeiAOProtectedChannel*>(aoSession.GetChannel(ch));
pAOChan->SetADCChannel(0, UeiAODiagnosticCurrent);

// Set ADC channel 1 to monitor voltage

pAOChan->SetADCChannel(1, UeiAODiagnosticVoltage);

// Disable ADC channels 2, 3 and 4

pAOChan->SetADCChannel(2, UeiAODiagnosticNone);

pAOChan->SetADCChannel(3, UeiAODiagnosticNone);
pAOChan->SetADCChannel(4, UeiAODiagnosticNone);
```

Enable circuit breaker channels: enable or disable circuit breaker; when enabled, monitor a diagnostic channel and open the circuit if any of the measurements go out of min/max limits:

```
// Enable CB on ADCChannel 0

pAOChan->EnableCircuitBreaker(0, true);

// Enable CB ADCChannel 1

pAOChan->EnableCircuitBreaker(1, true);
```

Set low/high limits: each circuit breaker can monitor one diagnostic channel. Set the minimum and maximum current, voltage, or temperature on the specified breaker. The circuit will open when any of the diagnostic values fall below or above those limits:

```
// Set CB channel 0 low limit for diagnostic ADCChannel 0
pAOChan->SetCircuitBreakerLowLimit(0,-0.01);
// Set CB channel 0 high limit
pAOChan->SetCircuitBreakerHighLimit(0, 0.01);
```

The three diagnostic measurements are set by default to:

	AO-318	AO-318-020	ADC Range
Electrical Current	±11 mA	-1 to 21 mA	±32mA
Electrical Voltage	±11 V	-1 to 16 V	±16V
Temperature	-40 to 85 °C		-50 to 125°C

Set over/under limit count: Specifies the number of consecutive over/under limit diagnostic readings that must occur in order to trip the circuit breaker. Can be set to trip the breaker on the initial read (0) or for up to 15 retries (~15 seconds).

```
// Set over/under count to 5 (valid range is 0 to 15)
pAOChan->SetOverUnderCount(5);
```

2.8.2 Status

The circuit breaker status can be accessed with a CUeiCircuitBreaker object.

```
// Create a circuit breaker and link it to the session’s stream for
//   circuit breaker on analog output channel (channel 4 in this case)
CUeiCircuitBreaker cb(aoSession.GetDataStream(), 4);
```

Call the `ReadStatus` method to retrieve the status masks.

Each bit in the current status mask corresponds to a circuit breaker. 1 if CB is currently tripped, 0 otherwise.

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

```
// Read CB status
uInt32 currStatus, stickyStatus
cb.ReadStatus(&currStatus, &stickyStatus);
```

2.8.3 Manual Reset

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

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

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

2.9 Cleaning-up the Session

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

```
// clean up the sessions
aiSession.CleanUp();
aoSession.CleanUp();
```

Chapter 3 Programming with the Low-level API

This chapter provides the following information about programming the AO-318 using the low-level API:

- About the Low-level API (Section 3.1)
- Low-level Functions (Section 3.2)
- Low-level Programming Techniques (Section 3.3)
  - Data Transfer Modes (Section 3.3.1)
- Programming the AO-318 (Immediate Mode) (Section 3.4)
  - Configuring Output Channels (Section 3.4.1)
  - Writing Output Data (Section 3.4.2)
  - Configuring Guardian Diagnostics (Section 3.4.3)
  - Monitoring Diagnostic Measurements (Section 3.4.4)
  - Configuring Circuit Breakers (Section 3.4.5)
  - Reading Circuit Breaker Status (Section 3.4.6)
  - Reengaging Circuit Breakers (Section 3.4.7)

3.1 About the Low-level API

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

When programming in Windows OS, however, we recommend that you use the UeiDaq high-level Framework API (see **Chapter 2**). The Framework simplifies the low-level API that makes 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 systems:
  <PowerDNA-x.y.z>/docs
- On Windows systems:
  Start » All Programs » UEI » PowerDNA » Documentation

3.2 Low-level Functions

Table 3-1 provides a summary of AO-318-specific functions. All low-level functions are described in detail in the PowerDNA API Reference Manual.

Table 3-1 Summary of Low-level API Functions for DNx-AO-318

Function	Description
`DqAdv3xxWrite`	Write either floating point or raw values to AO-318 output.
`DqAdv318ReadADC`	Read back voltage, current, temperature from Guardian diagnostic ADCs.
`DqAdv318Reengage`	Reset tripped circuit breakers for selected channels.
`DqAdv318CBStatus`	Get the circuit breaker status for selected channels.
`DqAdv318SetCBLevels`	Configure circuit breaker minimum and maximum trip levels for selected channels.
`DqAdv318SetConfig`	Advanced configuration of circuit breakers and ADCs.

NOTE: The AO-318-024 is supported in PowerDNA version 4.10.0.48+. If you're unsure if your version supports the board please contact Technical Support at support@ueidaq.com

3.3 Low-level Programming Techniques

Application developers are encouraged to explore existing source code examples when first programming the AO-318. Sample code provided with the installation is self-documented and serves as a good starting point.

Code examples are located in the following directories:

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

Sample code has the name of the I/O boards being programmed embedded in the sample name. For example, SampleVMap3xx contains sample code for running an AO-318 (and other AO-3xx products) in VMap mode. Refer the low-level sample code `Sample318` for an example of how to use the Guardian diagnostic features, and `Sample318CB` for an example of how to use the Circuit Breaker functions.

3.3.1 Data Transfer Modes

The AO-318 supports the following acquisition modes.

Immediate (point-to-point): Transfers a single data point per channel of a single I/O board at a non-deterministic pace.
Runs at a maximum of 100 Hz.

RtDMap/RtVMap: Transfers samples as specified in a user-defined map of I/O boards and channels. The timebase is maintained by the host application. Designed for closed-loop (control) applications.

- RtDMap delivers 1 data sample per channel
- RtVMap delivers multiple samples per channel

ACB: Transfers a buffer of samples, where transfers are initiated by IOM firmware issuing event notifications when circular buffer crosses a user-defined frame boundary. Every data point is guaranteed but will have an intrinsic delivery delay. ACB is not appropriate for control applications and is not supported for UEIPAC products.

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

3.4 Programming the AO-318 (Immediate Mode)

The following sections provide an overview of how to set up and use your AO-318 in Immediate Mode using the low-level API.

For best results, use this overview in conjunction with actual sample code, (e.g., Sample30x, Sample318, Sample318CB). This overview does not address all initialization or error handling. Refer to Section 3.3 for sample code location.

3.4.1 Configuring Output Channels

Users initialize a list of AO-318 channels to enable and output samples to.

```
uint32 cl[CHANNELS]; // CHANNELS is max of 8
```

You can enable channels sequentially or in whichever order you choose:

```
// to order channels sequentially in the channel list:
for (i = 0; i < CHANNELS; i++) {
      cl[i] = i;
}
```

3.4.2 Writing Output Data

In Immediate mode, use the `DqAdv3xxWrite()` API to write raw or floating point samples to each of the enabled channels (in the order of the channel list):

```
uint16 data[CHANNELS];
double fdata[CHANNELS];
```

```
while (!stop) {

    //set up data to be output
    for (i = 0; i < CHANNELS; i++) {
          data[i] = 0; // value isn't used, not sending raw data
          fdata[i] = nextSampleToOutput();   // -10V through 10V for AO-318
                                             //  0 to 20 mA for AO-318-020
    }

    DqAdv3xxWrite(hd,        // handle to IOM
                  DEVN,      // position AO-318 inserted in the chassis
                  CHANNELS,  // total number of channels enabled
                  cl,        // channel list configured in previous step
                  0,         // RawData flag (0 is FALSE, use floating pt)
                  data,      // array of raw data - not used if RawData==0
                  fdata);    // CHANNELS-size array of floating point data

    UeiPalSleep(500); // controls data output rate

}
```

3.4.3 Configuring Guardian Diagnostics

When accessing Guardian features, users configure which of the five AO-318 diagnostic points to monitor for each output channel.

Diagnostic voltages, current, and temperature are defined as follows:

Diagnostic ADC Channel	AO-318 Description (see **Figure 3-1**)	AO-318-02x Description (see **Figure 3-2**)
(0)	Iin current	Iin current
(1)	DACAOut buffer voltage	IntDAC buffer voltage
(2)	DACBOutB buffer voltage	IntDACR pre-relay voltage
(3)	AOutAB voltage at the connector pin	AOutA voltage at the connector pin
(4)	Temperature	Temperature

Table 3-2. AO-318 Diagnostic Measurements

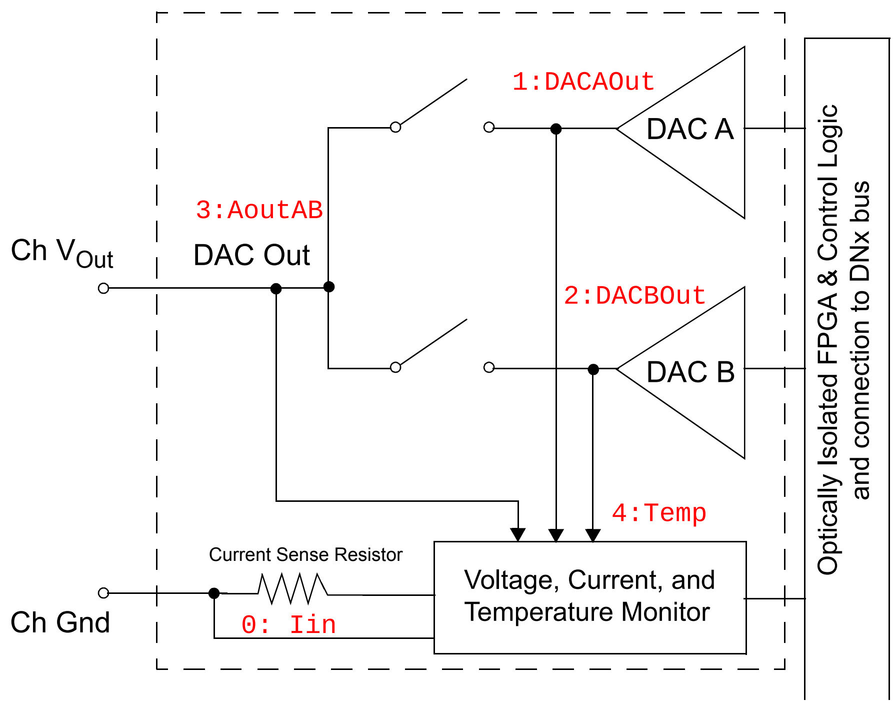

Figure 3-1 Block Diagram of Diagnostic Measure Points for AO-318 Channel

NOTE: The AO-318-02x current readings are valid within a few microamps. The voltage reading may read lower because of the voltage drop across the DNA-CBL-37 or similar cable which is not accounted for in the voltage readings.

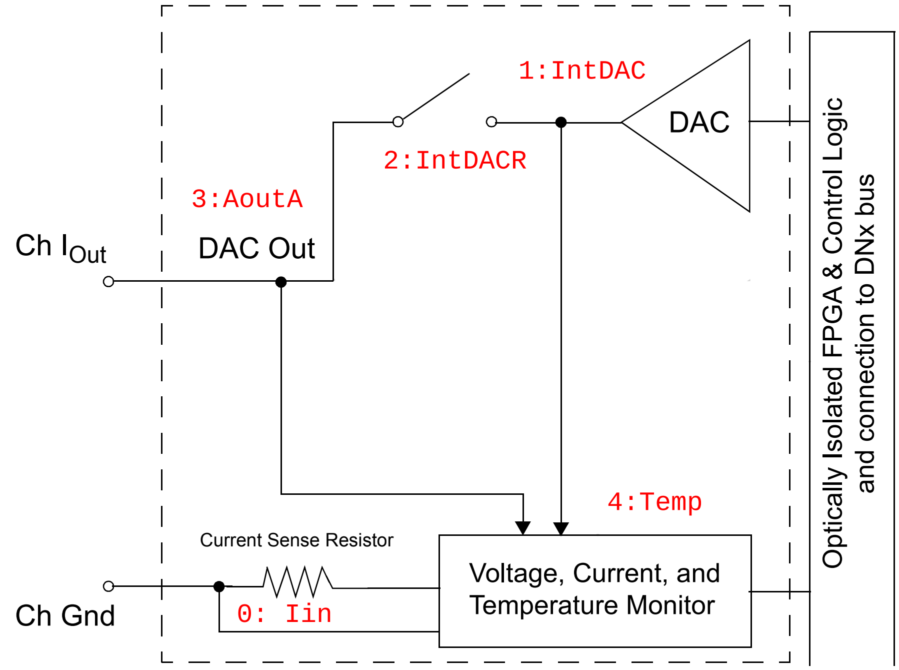

Figure 3-2 Block Diagram of Diagnostic Measure Points on an AO-318-02x Channel

Users set up a read channel list with an array size of the total number of diagnostic data values that will be read:

```
// RCHANNELS represents all diagnostic channels to read (and optional
//      timestamp)

//  In this example, all 5 ADC channels (diagnostic data points) are
//  configured for each of the enabled output channels (CHANNELS)

#define TIMESTAMP (0)            // timestamp is not included for this example
#define RCHANNELS ((5 * CHANNELS)+TIMESTAMP) // read all 5 diagnostic points

uint32 rcl[RCHANNELS];      // create a channel list of readback data
```

To create entries for the read channel list, diagnostic channels (as listed in Table 3-2) are logically ORed with the output channel number using the UEI macro, `DQ_AO318_MAKE_CL(OutputChannel,DiagnosticChannel)`:

```
for (i = 0; i < CHANNELS; i++) {
    rcl[i*5+0] = DQ_AO318_MAKE_CL(i,0);     // read diagnostic Iin
    rcl[i*5+1] = DQ_AO318_MAKE_CL(i,1);     // read diagnostic DACAOut
    rcl[i*5+2] = DQ_AO318_MAKE_CL(i,2);     // read diagnostic DACBOut
    rcl[i*5+3] = DQ_AO318_MAKE_CL(i,3);     // read diagnostic AoutAB
    rcl[i*5+4] = DQ_AO318_MAKE_CL(i,4);     // read diagnostic Temp
}
```

3.4.4 Monitoring Diagnostic Measurements

Call the `DqAdv318ReadADC()` API to program the ADC hardware, and then successive `DqAdv318ReadADC()` calls read the diagnostic data values (in the order of the read channel list):

```
// The first read is for hardware configuration only. Pass zeros instead
//    of arrays since no data is returned

DqAdv318ReadADC(hd,        // handle to IOM
               DEVN,      // position AO-318 inserted in the chassis
               RCHANNELS, // total number of input (diagnostic channels)
               rcl,       // read channel list configured in previous step
               0,         // set to 0 for config
               0);        // set to 0 for config

uint32 b_adcdata[CHANNELS*5 + TIMESTAMP];
double f_adcdata[CHANNELS*5 + TIMESTAMP];

DqAdv318ReadADC(hd,        // handle to IOM
               DEVN,      // position AO-318 inserted in the chassis
               RCHANNELS, // total number of diagnostic channels enabled
               rcl,       // read channel list
               b_adcdata, // array for returned raw readback data
               f_adcdata);// array for returned floating pt readback
```

3.4.5 Configuring Circuit Breakers

Each output channel on the AO-318-1 has 2 circuit breakers, and each output on the AO-318-02x has one.

These are configured with the `DqAdv318SetConfig()` API and minimum/ maximum levels can be set with the `DqAdv318SetCBLevels()` API.

Circuit breakers have the following user programmable features:

- Each circuit breaker can be controlled by up to 2 sources that are user programmable.
- You can select 0, 1, or 2 of the 5 diagnostic channels (see **Table 3-2**) as a control source to monitor for a circuit breaker.
- Users can program a minimum or maximum value for any of the selected diagnostic channels as a control condition for the circuit breaker (or the min/max rule can be disabled to select 0).

In this example, we will configure one circuit breaker with only 1 control source. For that control source, we program a minimum and maximum trip point for the measurement:

1. Program circuit breaker on DACA (see **Figure 3-1**) on channel 0
2. Set the trip points to when the DACA output voltage is less than a -11.0 V minimum and greater than an 8.0 V maximum. (Diagnostic channel (1) is “DACAOut voltage” in this example)

NOTE: The following example uses many predefined #defined variables to configure functionality. The variables used are explained in the example. To learn more, please refer to the PowerDNA API Reference Manual.

3.4.5.1 Declare CB Configuration Structures

To configure the circuit breaker, first declare Circuit Breaker configuration structures.

```
DQAO318CFG cfg318; // used by DqAdv318SetConfig()
DQAO318BRK_CFG cb_cfg; // used by DqAdv318SetCBLevels()
```

3.4.5.2 Program CB Enable & Criteria Selection

Set the `DQAO318CFG` structure to enable the circuit breaker and to use the DACA output voltage as the control criteria.

Unmask the programming of the DAC enable, source mode, and ADC channel list:

```
cfg318.prmmask = DQ_AO318_CBCFG_DAC |
                 DQ_AO318_CBCFG_SETCBSRC |
                 DQ_AO318_CBCFG_SETADCCL;
```

Enable DACA:

```
cfg318.en_DAC   =   DQAO318CFG_DACEN_A; // enable CB_A
```

Select a source as the control for CB_A; In this example, we only enable one source and disable both on CB_B (CB_B is the CB on the redundant DAC):

```
cfg318.CB_mode  =  DQ_AO318_CB_A_SEL0(DQ_AO318_MINMAX_1) |
                    DQ_AO318_CB_A_SEL1(DQ_AO318_MINMAX_DISABLE)|
                    DQ_AO318_CB_B_SEL0(DQ_AO318_MINMAX_DISABLE)|
                    DQ_AO318_CB_B_SEL1(DQ_AO318_MINMAX_DISABLE)|
                   0;
```

Map the ADC diagnostic channels to `ADC_CL[]`, the circuit breaker ADC channel list.

This example sets `CB_mode` to monitor `ADC_CL[1]`, `DQ_AO318_MINMAX_1`. This causes the diagnostic ADC channel measurement that is programmed as the `ADC_CL[1]` element to be selected as the min/max control criteria. (If we had programmed `DQ_AO318_CB_A_SEL0` with `DQ_AO318_MINMAX_0`, then `ADC_CL[0]` (Iin) could be the control criteria).

NOTE: For the AO-318, ONLY the first 3 ADC_CL elements can be used as a control criteria: put whichever values you want to monitor in the first 3 locations.

```
cfg318.ADC_CL[0] = DQ_AO318_ADC_CH_I;     //DQ_AO318_MINMAX_0 will be Iin
cfg318.ADC_CL[1] = DQ_AO318_ADC_CH_A_int; // set ADC_CL[1] to DACAOut,
                                          //   DQ_AO318_MINMAX_1 measurement
cfg318.ADC_CL[2] = DQ_AO318_ADC_CH_TEMP;  //DQ_AO318_MINMAX_2 is temperature
cfg318.ADC_CL[3] = DQ_AO318_ADC_CH_AB_ext; // Vout (AB) is not used
cfg318.ADC_CL[4] = DQ_AO318_ADC_CH_B_int;  // DACBOut is not used
```

Optionally, set the ADC sample rate (50 Hz max) **and read count** (which controls the number of readings outside the min/max that must occur before the circuit breaker trips):

```
cfg318.ADC_rate = 10.0; // do not sample > 50 Hz;
cfg318.rdcnt = 2;       // 0 to 15 (0 corresponds to 1 reading)
```

Call the `DqAdv318SetConfig` API to set configuration:

```
int mask = 0x1;   // channel mask; 0x1 is 1st channel; 0xff is all 8 ch
DqAdv318SetConfig(hd, DEVN, mask, 0, &cfg318);
```

3.4.5.3 Program CB Minimum & Maximum Levels

Set the DQAO318BRK_CFG structure to program the minimum and maximum levels for the diagnostic channels you programmed in ADC_CL[0], ADC_CL[1], and ADC_CL[2].

NOTE: Note that what you select for CB_mode is what ADC diagnostic channel will actually be read and compared to its min/max as the criteria for tripping the circuit breaker. In our example, only ADC_CL[1] is selected in CB_mode; the other two (Iin and Temperature) will not be criteria.

Program min/max values:

```
cb_cfg.CB_val_min_f[0] = (float)-0.011;     // A: ADC_CL[0] is Iin, not used
cb_cfg.CB_val_max_f[0] = (float) 0.011;     // A
cb_cfg.CB_val_min_f[1] = (float)-11.0;      // min V: ADC_CL[1] is criteria
cb_cfg.CB_val_max_f[1] = (float) 8.0;       // max V
cb_cfg.CB_val_min_f[2] = (float)-45.0;      // deg C: ADC_CL[2] not used
cb_cfg.CB_val_max_f[2] = (float) 85.0;      // deg C
```

Program units:

```
cb_cfg.units[0] = 'I';      // this is 'I' because cfg318.ADC_CL[0] is set
                            //     to read the current Iin as ADC input
cb_cfg.units[1] = 'V';      // this is 'V' because cfg318.ADC_CL[1] is set
                            //     to read DACBOut voltage as ADC input
cb_cfg.units[2] = 'T';      // this is 'V' because cfg318.ADC_CL[2] is set
                            //     to read the temperature as ADC input
```

Call the DqAdv318SetCBLevels API to set min/max levels:

```
// channel mask; 0x1 is 1st channel; 0xff is all 8 ch
DqAdv318SetCBLevels(hd, DEVN, mask, &cb_cfg);
```

3.4.6 Reading Circuit Breaker Status

To monitor the status of the circuit breakers, use the DqAdv318CBStatus() API:

```
// declare an array to hold status states for board and all channels
//   cbstatus[0] is bitmask of all channels; cbstatus[1:8] are individual ch
uint32 cbstatus[DQ_AO318_CHAN+1]; // DQ_AO318_CHAN is all 8 channels

// channel mask; 0x1 is 1st channel; 0xff is all 8 ch
DqAdv318CBStatus(hd, DEVN, mask, 0, 0, cbstatus);
```

- cbstatus[0]:
  bit 23:16 - current status of CB for ch 7:0 (1 is tripped)
  (bits 15:0 are sticky; cleared on read)
  bit 15:8 - indicates ADC min/max value has been evaluated for ch 7:0
  bit 7:0 - sticky indicator that ch 7:0 CB had been tripped since last read
- cbstatus[1:8]:
  refer to PowerDNA API Reference Manual

3.4.7 Reengaging Circuit Breakers

You can configure the circuit breakers to auto reengage when setting up the configuration with the `DqAdv318SetConfig` API, or you can monitor the circuit breaker status, and reengage circuit breakers with the API:

3.4.7.1 Automatically Reengaging CB

When you set up the `CB_mode` for the `DqAdv318SetConfig` API (see Section 3.4.5.2), you can OR in a `DQ_AO318_CB_A_AUTO` or `DQ_AO318_CB_B_AUTO` flag (or both) that enables auto reengaging on the DACA circuit breaker or DACB circuit breaker:

```
cfg318.CB_mode  =  DQ_AO318_CB_A_SEL0(DQ_AO318_MINMAX_1) |
                    DQ_AO318_CB_A_SEL1(DQ_AO318_MINMAX_DISABLE)|
                    DQ_AO318_CB_B_SEL0(DQ_AO318_MINMAX_DISABLE)|
                    DQ_AO318_CB_B_SEL1(DQ_AO318_MINMAX_DISABLE)|
                   0;
// Optionally, OR in auto reengage flag
cfg318.CB_mode |= (DQ_AO318_CB_A_AUTO|DQ_AO318_CB_B_AUTO);
```

3.4.7.2 Manually Reengaging CB

If you do not want to automatically reengage circuit breakers, you can call the `DqAdv318Reengage` API upon your reengagement criteria.

In this example, we assume the status has been read with the `DqAdv318CBStatus` API (Section 3.4.6), and we are reengaging all of the circuit breakers if the criteria is met and any of the circuit breakers have tripped.

```
// pass the DqAdv318Reengage API a bitmask of channels to reengage
uint32 resetAllChannels = 0xff;
if ((yourCriteria == TRUE) && ((cbstatus[0]>>16) & 0xff)){
            DqAdv318Reengage(hd, DEVN, resetAllChannels);
}
```

Appendix A

A.1 Accessories

The following cables and STP boards are available for the AO-318 board.

DNA-CBL-37

This is a 37-conductor flat ribbon cable with 37-pin male D-sub connectors on both ends. The length is 3ft and the weight is 3.4 ounces or 98 grams.

DNA-CBL-37S

This is a 37-conductor round shielded cable with 37-pin male D-sub connectors on both ends. It is made with round, heavy-shielded cable; 3 ft (90 cm) long, weight of 10 ounces or 282 grams; also available in 10ft and 20ft lengths.

DNA-STP-37

The DNA-STP-37 provides easy screw terminal connections for all DNx 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 cable. The dimensions of the STP-37 board are 4.2w x 2.8d x1.0h inch or 10.6 x 7.1 x 7.6 cm (with standoffs). The weight of the STP-37 board is 2.4 ounces or 69 grams.

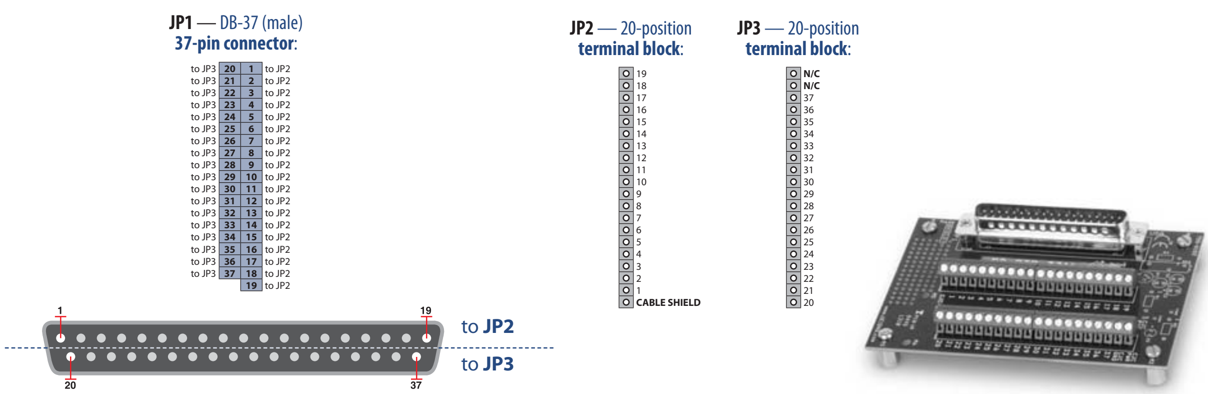

Figure A-1 Pinout and photo of DNA-STP-37 screw terminal panel

Index