



DNx-AO-308 User Manual

**8-Channel, 16-bit, $\pm 10V$ Analog Output Board
for the PowerDNA Cube and RACK series chassis**

August 2021

PN Man-DNx-AO-308

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
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

Chapter 1 Introduction	1
1.1 Organization of Manual	1
1.2 AO-308 Board Overview	3
1.2.1 DAC Output Capabilities	3
1.2.2 Isolation & I/O Connections	3
1.2.3 Environmental Conditions	3
1.2.4 Software Support	3
1.3 Features	4
1.4 Specification	4
1.5 Device Architecture	5
1.6 Indicators	5
1.7 Wiring & Connections (pinout)	6
Chapter 2 Programming with the High-Level API	7
2.1 About the High-level Framework	7
2.2 Creating a Session	7
2.3 Configuring the Resource String	7
2.4 Configuring Channels for Output	8
2.5 Configuring the Timing	8
2.6 Writing Data	8
2.7 Cleaning-up the Session	9
Chapter 3 Programming with the Low-level API	10
3.1 About the Low-level API	10
3.2 Low-level Functions	10
3.3 Low-level Programming Techniques	11
3.4 Data Output Modes	12
3.4.1 Immediate (Point-by-Point) Mode	12
3.4.2 Waveform Re-generation	12
3.4.3 DMAP Mode	12
3.4.4 VMAP Mode	12
3.4.5 ACB Mode	12
3.5 Programming the AO-308-xxx (Immediate Mode)	13
3.5.1 Setting up Channel Configuration	13
3.5.2 Writing Output Data	13
3.6 Configuration Settings	14
3.7 Data Representation	15



List of Figures

Chapter 1 Introduction	1
1-1 Block Diagram of the AO-308 Layer	5
1-2 Photo of DNA-AO-308 Layer	5
1-3 Pinout Diagram of the AO-308 Layer	6
Chapter 2 Programming with the High-Level API	7
Chapter 3 Programming with the Low-level API	10
A-1 Pinout and photo of DNA-STP-37 screw terminal panel.....	16



List of Tables

Chapter 1 Introduction	1
1-1 DNx-AO-308 Technical Specifications.....	4
Chapter 2 Programming with the High-Level API	7
Chapter 3 Programming with the Low-level API	10
3-1 Summary of Low-level API Functions for DNx-AO-308-xxx.....	10



Chapter 1 Introduction

This document outlines the feature set and use of the DNx-AO-308, 8-channel digital-to-analog output boards.

The following sections are provided in this chapter:

- Organization of Manual (Section 1.1)
- AO-308 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)

NOTE: Note that the following DNx-AO-308-xxx series boards are described in separate manuals:

- AO-308-350 16-bit, 8 channel, ± 10 V, High Current Analog Output Board
- AO-308-352 16-bit, 8 channel, ± 13.5 V, Medium Voltage/Current Analog Output Board
- AO-308-353 16-bit, 8 channel, ± 40 V, High Voltage Analog Output Board
- AO-308-020 16-bit, 8 channel, 0-20 mA Current Analog Output Board
- AO-308-420 16-bit, 8 channel, 4-20 mA Current Analog Output Board

Please refer to those separate manuals for more information.

1.1 Organization of Manual

This AO-308 User Manual is organized as follows:

- **Introduction**
Chapter 1 provides an overview of DNx-AO-308 Analog Output Board 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 generate output on the DNx-AO-308 with the UEIDAQ High-level Framework API.
- **Programming with the Low-Level API**
Chapter 3 is an overview of low-level API commands for configuring and using the AO-308 series I/O board.
- **Appendix A - Accessories**
This appendix provides a list of accessories available for use with the DNx-AO-308 board.

NOTE: A glossary of terms used with the PowerDNA Cube/Rack and layers can be viewed and/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-308 Board Overview** The DNx-AO-308 are high-precision, true 16-bit, 8-channel voltage analog output boards.
- DNA-AO-308, DNR-AO-308, and DNF-AO-308 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 per-channel digital offset and gain calibration, buffered output, excellent linearity, and low output noise. DNx-AO-308 layer is capable of outputting ± 10 V with maximum current of ± 5 mA per channel. This board is recommended for output applications that draw less than ± 5 mA. For applications that require higher output current, use DNx-AO-308-350 layer.
- 1.2.2 Isolation & I/O Connections** The outputs of the DNx-AO-308 are fully isolated from 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-308 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.
- 1.2.3 Environmental Conditions** As with all UEI PowerDNA boards, the DNx-AO-308 can be operated in harsh environments and has been tested at 5g vibration, 50g shock, -40 to +85°C temperature, and altitudes up to 70,000 feet or 21,000 meters.
- 1.2.4 Software Support** Software included with the DNx-AO-308 provides a comprehensive yet easy to use API that supports Windows, Linux, and real-time operating systems, such as QNX, VxWorks and more. 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-308 layer has the following features:

- 8 independent DACs with 16-bit resolution
- 100 kHz per channel max update rate
- ± 10 V output range, ± 5 mA per channel
- Glitch-free output
- Per-channel offset and gain calibration
- Simultaneous update across all channels
- Tested to withstand 5 g vibration, 50 g shock, -40 to +85 °C temperature, and altitude up to 70,000 ft or 21,000 meters.

1.4 Specification

The technical specification for the DNx-AO-308 board are listed in **Table 1-1**.

Table 1-1 DNx-AO-308 Technical Specifications

Number of Channels	8
Resolution	16 bits
Max Update Rate: @ 16-bit resolution	100 kHz/channel (500kHz max aggregate)
Buffer Size	1K samples
Type of D/A	double-buffered
INL (no load)	± 1 LSB (0.003%)
DNL (no load)	± 1 LSB (0.003%)
Monotonicity Over Temperature	16 bits
Gain Linearity Error	0.002%
Gain Calibration Error	± 100 μ V
Offset Calibration Error	± 100 μ V
Offset Drift	± 10 μ V/ $^{\circ}$ C
Gain Drift	5ppm/ $^{\circ}$ C
Output Range	± 10 V
Output Coupling	DC
Output Impedance	0.1 Ω max
Current Drive	± 5 mA/channel
Capacitive Loads	500 pF
Settling Time	10 μ s to 16 bits
Slew Rate	10 V/ μ s
Isolation	350Vrms
Power Consumption	1.5W - 3W
Physical Dimensions	3.875" x 3.875" (98 x 98 mm)
Operating Temp. (tested)	-40 to 85 deg C
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

NOTE: The AO-308 does not have separate force and sense lines.



1.5 Device Architecture

Figure 1-1 is a block diagram of the architecture of the AO-308 layer.

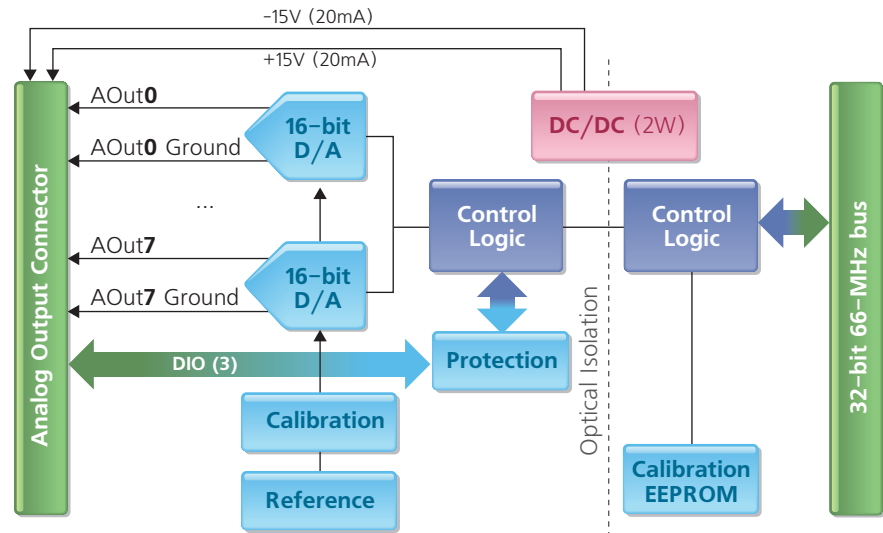


Figure 1-1. Block Diagram of the AO-308 Layer

The DNx-AO-308 Analog Output board has eight individual analog output channels. Each channel has its own A/D converter for the diagnostic. The board is optically isolated from the others.

1.6 Indicators

A photo of the DNx-AO-308 unit is shown in Figure 1-2.

The front panel has two LED indicators:

- RDY: indicates that the layer is receiving power and operational.
- STS: can be set by the user using the low-level framework.

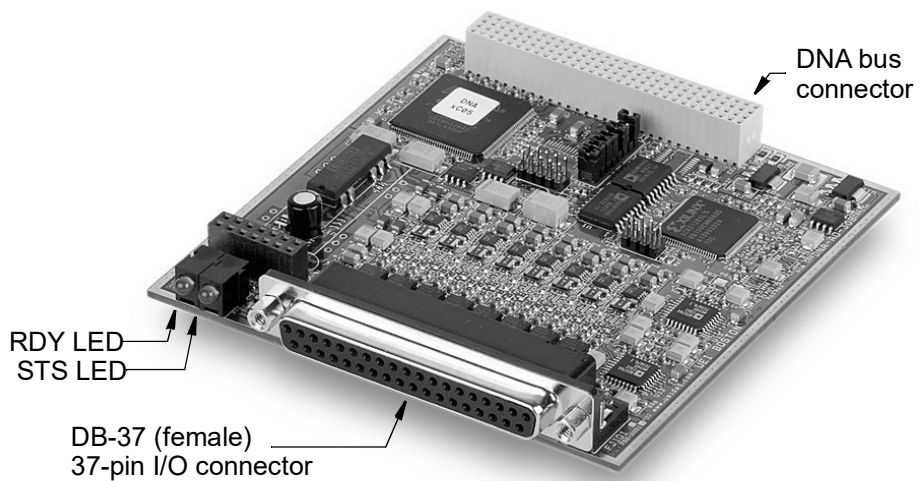


Figure 1-2. Photo of DNA-AO-308 Layer

1.7 Wiring & Connections (pinout)

Figure 1-3 below illustrates the pinout of the AO-308.

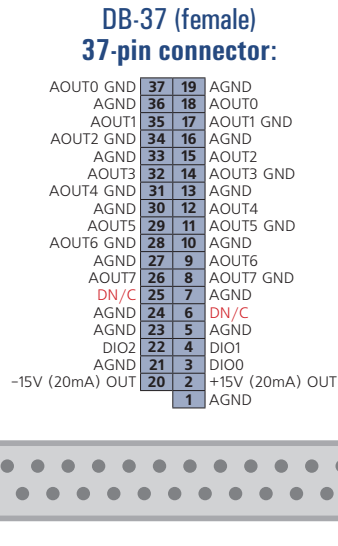


Figure 1-3. Pinout Diagram of the AO-308 Layer

NOTE: Do not connect to any pins marked as DN/C.



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-308:

- About the High-level Framework (Section 2.1)
- Creating a Session (Section 2.2)
- Configuring the Resource String (Section 2.3)
- Configuring Channels for Output (Section 2.4)
- Configuring the Timing (Section 2.5)
- Writing Data (Section 2.6)
- Cleaning-up the Session (Section 2.7)

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 input
CUEISession session;
```

2.3 Configuring the Resource String

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

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

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

For example, the following resource string selects 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 Channels for Output The AO-308 can be configured for analog output:

```
// Configure channels 0,1 with an output range of ±10V
session.CreateAOChannel("pdna://192.168.100.2/Dev0/ao0:1",
                        -10.0, 10.0);
```

2.5 Configuring the Timing You can configure the AO-308 to run in simple mode (point by point).
 In simple mode, the delay between samples is determined by software on the host computer.
 The following sample shows how to configure the simple mode. Please refer to the “UeiDaq Framework User’s Manual” to learn how to use other timing modes.

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

2.6 Writing Data Writing data is done using *writer* object(s).
 You can also create a writer object that writes data scaled to volts. 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(session.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-308-xxx analog output layers are programmed the same way.

2.7 Cleaning-up the Session

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

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



Chapter 3 Programming with the Low-level API

This chapter provides the following information about programming the DNx-AO-308 Analog Output Board using the low-level API. All AO-308-xxx layers are programmed the same way.

- About the Low-level API (Section 3.1)
- Low-level Functions (Section 3.2)
- Low-level Programming Techniques (Section 3.3)
- Data Output Modes (Section 3.4)
- Programming the AO-308-xxx (Immediate Mode) (Section 3.5)
- Configuration Settings (Section 3.6)
- Data Representation (Section 3.7)

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 or real-time operating systems.

When programming in Windows OS, however, we recommend that you use the UeiDaq high-level Framework API (see **Chapter 2**). The Framework extends the low-level API with additional functionality that makes programming easier and faster.

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:
C:\Program Files (x86)\UEI\PowerDNA\Documentation

3.2 Low-level Functions

Table 3-1 provides a summary of AO-308-xxx 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-308-xxx

Function	Description
DqAdv3xxWrite	Write either floating point or raw values to AO-308-xxx output.
DqAdv3xxSetWForm	Set configuration for continuous waveform output.
DqAdv3xxWriteWFormCL	Loads repetitive waveform into the analog output layer memory.
DqAdv3xxEnableWForm	Configures, enables, pause/resumes, or disables waveform output.



3.3 Low-level Programming Techniques

Application developers are encouraged to explore the existing source code examples when first programming the AO-308-xxx. 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: *C:\Program Files (x86)\UEI\PowerDNA\SDK\Examples*

Sample code for data acquisition modes have the name of the mode and the name of the I/O boards being programmed embedded in the sample name. For example, *SampleVMap3xx* contains sample code for running the an AO-308-xxx using VMap data acquisition mode. Note that immediate mode samples are named *Sample<I/O board name>*, (i.e., *Sample30x*).



- 3.4 Data Output Modes** The AO-308-xxx I/O board can be programmed using the following modes. You can only use one mode at a time and cannot, for example, run one channel in immediate mode and the other in waveform mode. API that implement data acquisition modes and additional mode descriptions are provided in the *PowerDNA API Reference Manual*.
- 3.4.1 Immediate (Point-by-Point) Mode** This is the easiest way to program the AO-308-xxx but also the least efficient. Immediate mode writes a single data point per channel to the layer at a non-deterministic pace. The maximum point-by-point rate is 100 Hz. Use the API `DqAdv3xxWrite()` to update the AO-308-xxx outputs with a DC voltage. The outputs will update on the next clock tick. When `DqAdv3xxWrite()` is called for the first time, the firmware terminates any ongoing operation on the device. Then, the firmware parses the channel list and writes the passed values one by one. This function cannot be called when the layer is involved in any streaming or data mapping operations. Every write to the channel takes approximately 3.3 μ s. Thus, execution time for this function depends on the number of channels in the channel list. AO-308-xxx runs its outputs at 1 kHz by default and you can change the update rate with `DqCmdSetClock()`. The example code `Sample30x` demonstrates immediate mode on AO-308-xxx.
- 3.4.2 Waveform Re-generation** This mode continuously generates a waveform uploaded in the AO-308-xxx's FIFO. The example code `Sample3xx_Waveform` demonstrates this mode.
- 3.4.3 DMAP Mode** Data Map (DMAP) mode is recommended to efficiently read and update one value per channel at a time. DMap is designed for closed-loop (control) applications and may include channels across multiple I/O layers. It guarantees that all I/O layers configured in the same DMAP are synchronized (all input samples are read at the same time and all output values are updated at the same time). The example code `SampleRTDMap3xx` demonstrates RTDMAP mode.
- 3.4.4 VMAP Mode** Variable Data Map (VMAP) mode reads and updates a variable number of data points per channel at a time. VMap is designed for closed-loop (control) applications and may include channels across multiple I/O layers. It guarantees that all I/O layers configured in the same VMAP are synchronized (all input samples are read at the same time and all output values are updated at the same time). The example code `SampleVMap3xx` demonstrates VMAP mode.
- 3.4.5 ACB Mode** Advanced Circular Buffer (ACB) mode continuously uploads and generates an arbitrary waveform to the AO-308-xxx outputs. It uses an event based software architecture to notify the user's when it's time to upload a new block of data. This is the only mode that can update the AO-308-xxx outputs at maximum speed. Not available on UEIPAC products. The example code `SampleACB30x` demonstrates ACB mode.



3.5 Programming the AO-308-xxx (Immediate Mode) The following sections provide an overview of how to set up and use your AO-308-xxx in Immediate Mode using the low-level API. For best results, use this overview in conjunction with actual sample code, (i.e., Sample30x). This overview does not address typical initialization or error handling. Refer to Section 3.3 for sample code location.

3.5.1 Setting up Channel Configuration Users initialize a list of AO-308-xxx channels to enable and output samples to. You can enable channels sequentially or in whichever order you choose

```
// CHANNELS is a max of 8.

uint32 cl[CHANNELS];

// Order channels sequentially in the channel list.

for (i = 0; i < CHANNELS; i++){
    cl[i] = i;
}
```

3.5.2 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];

// Set up and write output data.

while(!stop){
    for (i=0; i<CHANNELS; i++){
        data[i]=0; // Value isn't used, not sending raw data
        fdata[i]=nextSampleToOutput(); // see specification for output range
        DqAdv3xxWrite(hd, DEVN, CHANNELS, cl, 0, data, fdata);

        UeiPalSleep(500); // Controls data output rate
    }
}
```



3.6 Configuration Settings

Configuration settings are passed in `DqCmdSetCfg()` and `DqAcbInitOps()` functions.

Not all configuration bits apply to AO-308-xxx series layers. Some important definitions:

- `DQ_LN_ENABLE` enables all operations with the layer.
- `DQ_LN_CVCKSRC0` selects the internal channel list clock (CL) source as a timebase. AO-308 supports CV clock.
- `DQ_LN_ACTIVE` is needed to switch on “STS” LED on the CPU layer.

For example, for streaming operations with hardware clocking you can select the following flags:

```
DQ_LN_ENABLE | DQ_LN_CVCKSRC0 | DQ_LN_STREAMING |
DQ_LN_IRQEN | DQ_LN_ACTIVE
```

The AO-308-xxx also has a range of layer-specific settings, as follows:

```
// continuous output with FIFO. Mode reserved for future use.
#define DQ_AO3xx_MODEFIFO (1L << 19)
// waveform mode - continuous. Mode reserved for future use.
#define DQ_AO3xx_MODECONT (2L << 19)
// waveform mode - regenerate. Mode reserved for future use.
#define DQ_AO3xx_MODECYCLE (3L << 19)
// waveform mode - hardware. Mode reserved for future use.
#define DQ_AO3xx_MODEWFGEN (4L << 19)
```

You can select either the CL or CV clock as a timebase. Because of the parallel architecture of the AO-308-xxx layer, either clock triggers all converters.

$$\text{Aggregate rate} = \text{Per-channel rate} * \text{Number of channels}$$

Note that acquisition rate cannot be selected on per-channel basis.



3.7 Data Representation

AO-308-xxx has a 16-bit straight binary data representation:

Layer	Range/ Value	0x0	0x8000	0xFFFF	Span	Offset
AO-308, -350	±10V	-10V	0	+10V	20V	0
-352	±13.5V	-13.5V	0	+13.5V	27V	0
-353	±40V	-40V	0	+40V	80V	0
-020	0-20mA	0mA	10mA	20mA	20mA	0
-420	4-20mA	4mA	12mA	20mA	16mA	4mA

To convert voltage into an A/D representation, use the following formula:

$$Raw = (Volt+Offset) / (Span/0xFFFF) ,$$

where Volt is the desired level in volts.

To convert current into A/D representation, use the following formula:

$$Raw = (mA+Offset) / (Span/0xFFFF) ,$$

where mA is the desired level in milliamps.



Appendix A

A. Accessories

The following cables and STP boards are available for the AO-308 layer.

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 x 1.0h 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.

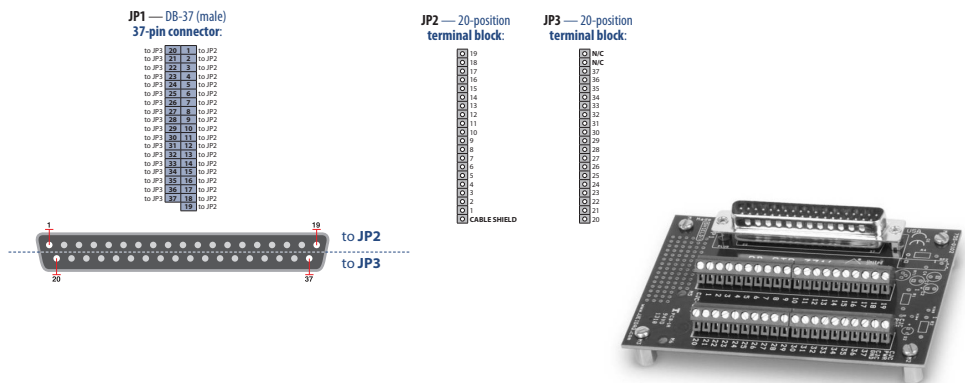


Figure A-1. Pinout and photo of DNA-STP-37 screw terminal panel

UEI-STP-AO-200

The UEI-STP-AO-200 is a unity gain, high current output buffer board for use with DNA-AO-308 and PD2/PDXI series analog output boards. It offers a gain of $1 \pm 0.2\%$ and drives up to 250 mA at ± 10 Volts. It requires a user-supplied ± 13 to ± 15 V power supply with current capability to support the output current specified. Users who do not want to supply bipolar power should order the UEI-STP-AO-200D unit, which includes a DC/DC converter and requires a +9V to +36V power supply.

NOTE: If the total power consumption of the layer exceeds 4.5W, a rear mount cooling fan such as the DNA-FAN5 (for 3-layer Cube) or DNA-FAN8 (for 5-layer Cube) should be added to the DNA Cube.

