United Electronic Industries

The High-Performance Alternative

DNA/DNR-AI-207 Analog Input Layer User Manual

Sequential Sampling, 18-bit, 16-channel, Differential Input Analog Input Layer with CJC

November 2013 Edition
Version 4.6
PN Man-DNx-AI-207-1113

© Copyright 1998-2010 United Electronic Industries, Inc. All rights reserved.

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 infringements 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 UEI's website for complete terms and conditions of sale:

http://www.ueidaq.com/company/terms.aspx

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 see

http://www.ueidaq.com/partners/

Support:

Telephone:(508) 921-4600

Fax:(508) 668-2350

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

Internet Support:

Supportsupport@ueidaq.com
Web-Sitewww.ueidaq.com
FTP Siteftp://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.

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

Table of Contents

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

Table of Figures

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

Chapter 1 Introduction

This document outlines the feature-set and describes the operation of the DNx-AI-207analog input boards. The DNA- version is designed for use with a PowerDNA Cube data acquisition system. The DNR- version is designed for use with a DNR-12 RACKtangle or a DNR-6 HalfRACK system. Both versions are functionally identical. Please ensure that you have the PowerDNA Software Suite installed before attempting to run examples.

1.1 Organization of this Manual

This PowerDNA AI-207 User Manual is organized as follows:

- **Introduction**
  Provides an overview of PowerDNA Analog Input Series board features, various models available, and what is needed to get started.
- **The AI-207 Layer**
  Provides an overview of the device architecture, connectivity, and logic of layer.
- **Programming Using the UeiDaq Framework High-Level API**
  Provides an overview of the how to create a session, configure the session for analog input, and interpret results on the AI-207 series layer.
- **Programming Using the Low-level API**
  Low-Level API commands for configuring and using the AI-207 series layer.
- **Appendix A: Accessories**
  This appendix provides a list of accessories available for AI-207 layer(s).
- **Index**
  This is an alphabetical listing of the topics covered in this 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 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.*

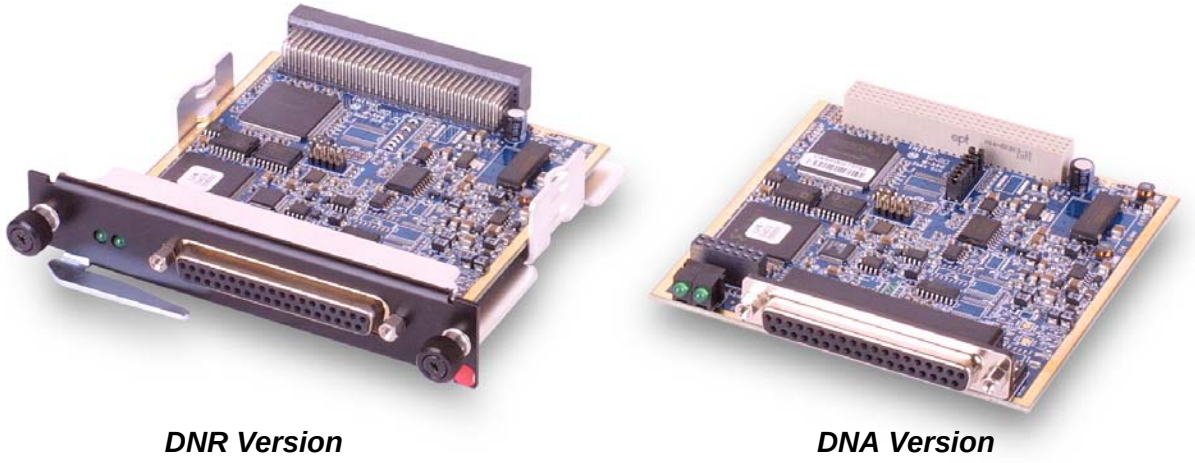
Text formatted in **bold** typeface generally represents text that should be entered verbatim. For instance, it can represent a command, as in the following example: “You can instruct users how to run setup using a command such as **setup.exe**.”

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

1.2 The AI-207 Layer

The AI-207 layer has the following features:

- 16 fully differential channels; additional dedicated CJC channel
- Maximum sampling rate of 1kHz per channel
- 18-bit resolution
- ±10V input range
- Gains: 1, 2, 4, 8, 10, 20, 40, 80, 100, 200, 400, 800
- Overvoltage protection (-40V to +55V)
- Dynamic autozero support
- Embedded averaging engine



DNR Version ***DNA Version***

Figure 1-1. Photos of DNA- and DNR-AI-207 Analog Input Layer Boards

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

1.3 Device Architecture

As shown in **Figure 1-2**, the DNX-AI-207 Layer has multiplexed inputs with a single 18-bit converter.

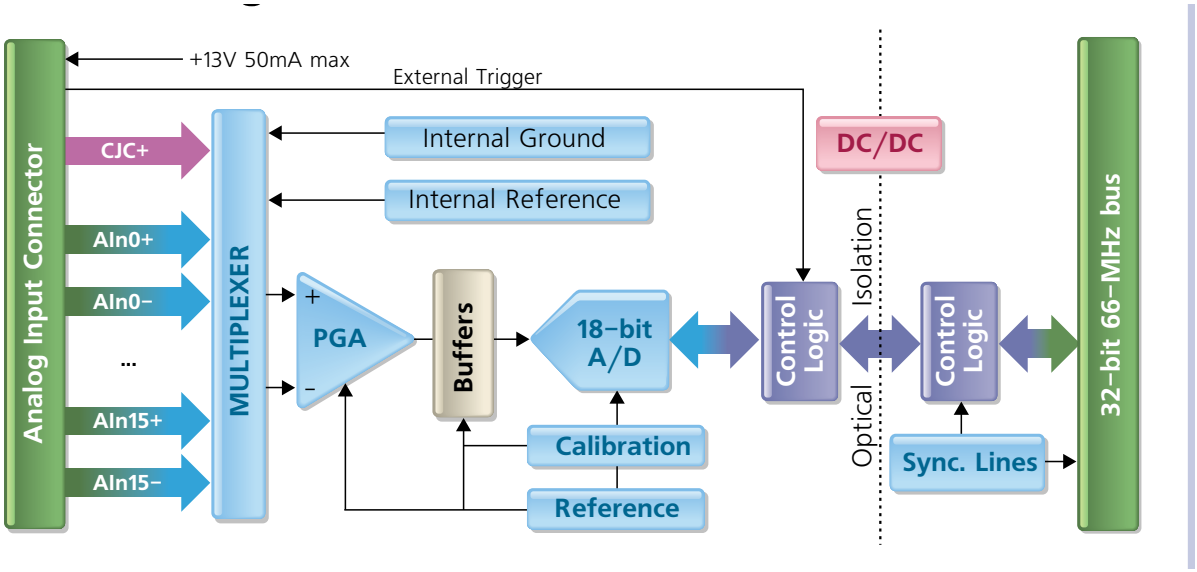


Figure 1-2. Block Diagram of DNx-AI-207 Layer Board

The DNx-AI-207 layer features 16 differential input channels with a maximum sampling rate of 1kS/s per channel, a wide range of gains, 18-bit resolution, and ±10V input range. Additionally, the DNx-AI-207 provides a dedicated CJC channel that can be used for reading from a built-in CJC sensor on the DNA-STP-AI-U accessory terminal panel. Static CJC compensation may be used when no CJC sensor is available (such as when the Layer is connected to a DNA-STP-37 accessory panel).

The DNx-AI-207 is designed for mid- to low-speed high-resolution signal measurement. Since this analog input layer achieves an exceptional cost/performance ratio, it represents an ideal solution for precise temperature measurement over a long period of time. When used with DNA-STP-AI-U Universal Terminal Panel, the DNx-AI-207 layer also offers a direct connection for thermocouples and open TC detection. The STP-AI-U accessory panel also provides excitation for 2-wire and 4-wire RTD measurements. For details, refer to the User Manual for the STP-AI-U screw terminal panel, which may be downloaded from www.ueidaq.com.

For a detailed description of how the UEI Framework software handles RTD measurements, refer to Chapter 2 of this manual.

One of the best features of the DNx-AI-207 is its automated offset autozero, which removes any offset fluctuations over the temperature range and/or time for every signal reading. This reduces temperature drift to a few microvolts over the full specified range. Another feature, the oversampling engine, permits the DNx-AI-207 to acquire as many samples as possible for the given gain/speed and automatically average them, dramatically improving noise rejection.

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

The DNx-AI-207 has an input multiplexer that sequentially selects each of the available input signals for input to a cascaded fully differential PGA and then to an 18-bit SAR converter. Logic on the isolated side controls channel switching, settling time delays, and the conversion process. Also, it reads data from the converter at the maximum possible rate and sends it over the isolation to the non-isolated logic for the further processing. The following additional channels, which are used to improve quality of the acquired signal, are internally connected to the multiplexers: internal ground, internal 2.5000V reference, and CJC channel.

When the channel list is processed by the PowerDNA firmware, every channel is allocated a required delay for best possible settling of the analog signal. The rest of the time is used for oversampling. The number of averages used in this computation is varied from 2 to 8096, depending on the selected acquisition rate and, because of the complexity of channel list processing, is set automatically by the firmware.

1.4 Layer Connectors and Wiring

The AI-207 layer uses a 37-pin female D-Sub connector with the following pinout:

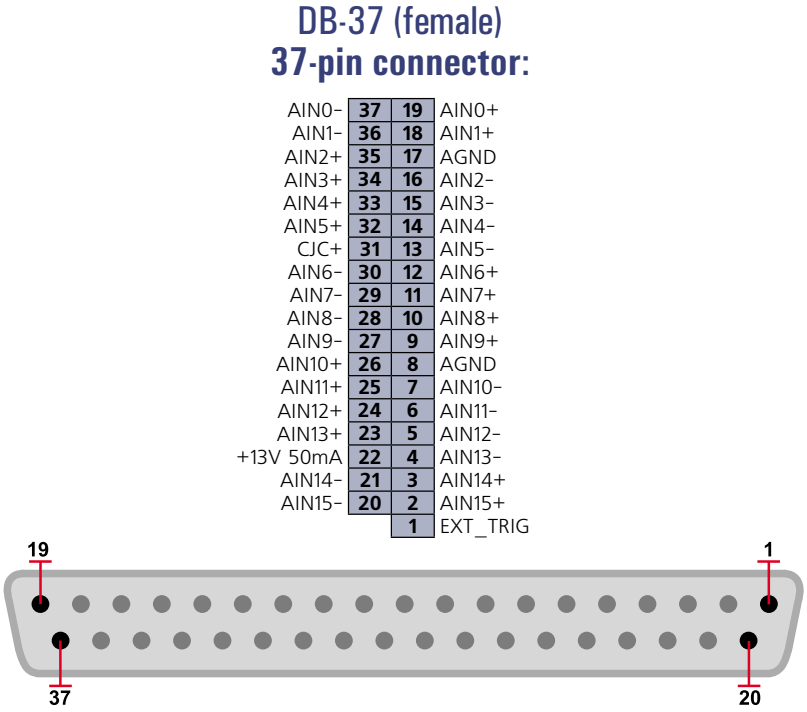


Figure 1-3. Pinout of the DNx-AI-207 Analog Input Layer

The AI-207 layer uses a B-size 37-pin D-sub connector. The following signals are located at the connector:

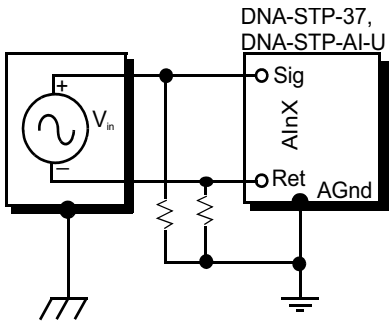
- AIN0+ – AIN15+ — input channel, differential mode.
  The AI-207 measures inputs between AIn+ and AIn- as long as common mode voltage is within limits. For single ended cnnections, AIn- should be connected to AGND.

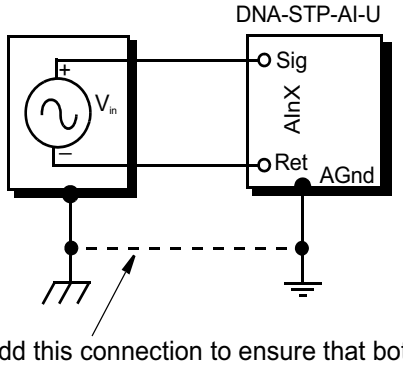
© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

- +13V 50mA – provides current.
- CJC+ — cold junction compensation return line from DNA-STP-AI-U.
- AGND — layer analog ground, isolated from system ground.
- EXT_TRIG — accepts an external trigger signal to the layer.

1.4.1 Analog Input Ground Connections

To avoid errors caused by common mode voltages on analog inputs, follow the recommended grounding guidelines in **Figure 1-4** below.

<table>
<tr><th rowspan="2">Input Configuration</th><th colspan="2">Type of Input</th></tr>
<tr><th>Floating</th><th>Grounded</th></tr>
<tr><td></td><td>Typical Signal Sources:
Thermocouples
DC Voltage Sources
Instruments or sensors with isolated outputs</td><td>Typical Signal Sources:
Instruments or sensors with non-isolated outputs</td></tr>
<tr><td>Differential</td><td> 

Two resistors (10k <R< 100k) provide return paths to ground for bias currents.</td><td> 

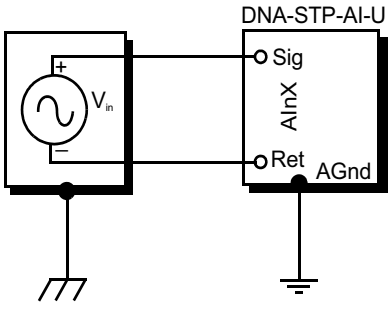
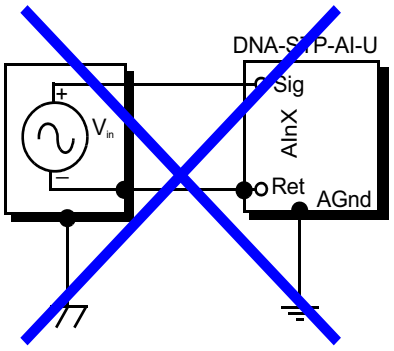
Add this connection to ensure that both grounds are at the same potential.</td></tr>
<tr><td>Single-Ended, Ground Referenced</td><td> 
</td><td>NOT RECOMMENDED
 
</td></tr>
</table>

Figure 1-4. Recommended Ground Connections for Analog Inputs

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

Because all analog input channels in AI-201/202/207/208/225 layers are isolated as a group, you can connect layer AGND to the ground of the signal source and eliminate the resistors shown in **Figure 1-4** for floating differential input signals.

1.5 Data Representation

The AI-207 layer is designed with 18-bit A/D converters. The AI-207 layer can return 18-bit straight binary data in 32-bit words.

The 18-bit data is represented as follows:

Bit	Name	Description	Reset State
17-0	ADCDATA	Upper 18 bits of data, straight binary	<pos>

<pos> represents a position in the output buffer. Upon reset, every entry in the output buffer is filled with its relative position number. As an initializing step, you should read the buffer and discard the data before proceeding with normal data collection.

If you start receiving consecutive data such as 0, 1, 2, from the layer after performing the anti-aliasing step, it means that the layer is either not initialized properly or the layer is damaged.

To convert data into floating point, use the following formula (at a gain of 1):

$$\text{Volts} = (\text{Raw} \wedge 3\text{FFFF}) \times \left(\frac{20\text{V}}{2^{18}}\right) - 10\text{V}$$

1.5.1 CJC Data

Raw CJC Voltage from the AI-207 may be represented as:

$$T_{\text{Kelvin}} = \frac{\text{CJCVoltage}}{0.00295}$$

For example, if the voltage read from Channel 33 (the CJC channel) is 0.87, the CJC Temperature is:

Temp. Scale	Calculation	CJC Temperature
Kelvin	0.87/0.00295	294.9 °K
Celsius	294.9 – 273.15	21.75 °C
Fahrenheit	1.8 x 21.75 + 32	71.75 °F

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

Chapter 2 Programming with the High Level API

This section describes how to program the DNx-AI-207 using the UeiDaq Framework API.

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, LabVIEW, or DASYLab.

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

Start » Programs » UEI » Framework » Examples

The following subsections focus on the C++ API, but the concept is the same regardless of programming language.

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

2.1 Creating a Session

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

```
CUeiSession session;
```

2.2 Configuring Channels and Excitation

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 such as:

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

For PowerDNA the device class is **pdna**.

2.2.1 Voltage Measurement

To program the analog input circuitry, configure the channel list using the session’s object method “CreateAIChannel”.

For example, the following resource string selects analog input channels 0,2,3,4 on device 1 at IP address 192.168.100.2: “pdna://192.168.100.2/Dev1/Ai0,2,3,4”

The gain applied on each channel is specified by using low and high input limits.

For example, the AI-207 available gains are 1, 2, 4, 8, 10, 20, 40, 80,100, 200, 400, 800 and the maximum input range is [-10V, 10V].

To select the gain of 100, you need to specify input limits of [-0.1V, 0.1V].

```
// Configure channels 0,1 to use a gain of 100 in differential mode
session.CreateAIChannel(“pdna://192.168.100.2/Dev0/Ai0,1”,
             -0.1, 0.1,
             UeiAIChannelInputModeDifferential);
```

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

2.2.2 Thermocouple Measurement

Thermocouple measurements are configured using the Session object's method "CreateTCChannel".
The measurements will be scaled in the unit specified by the "temperature scale" parameter.

Depending on your hardware, you can specify whether the scaling calculation will use a constant cold-junction temperature or whether you will measure it from a sensor built in the DNx-STP-AIU connector block

```
// Add 4 channels (0 to 3) to the channel list and configure
// them to measure a temperature between 0.0 and 1000.0 degrees C
// from type J thermocouples, scale temperatures in Celsius
// degrees and using CJC built-in compensation from the STP-AI-U,
// in differential mode.

MySession.CreateTCChannel("pdna://192.168.100.2/dev0/Ai0:3",
                          0, 1000.0,
                          UeiThermocoupleTypeJ,
                          UeiTemperatureScaleCelsius,
                          UeiColdJunctionCompensationBuiltIn,
                          25.0,
                          "",
                          UeiAIChannelInputModeDifferential);
 <!--[if !supportLineBreakNewLine]-->
 <!--[endif]-->
```

2.2.3 Resistance Measurement

Resistance measurements are configured using the Session object method "CreateResistanceChannel". To measure the resistance, we need to know the amount of current flowing through it. We can then calculate the resistance by dividing the measured voltage by the known excitation current. To measure the excitation current, we measure the voltage from a high precision reference resistor whose resistance is known. The reference resistor is built-into the terminal block if you are using a DNA-STP-AI-U, but you can provide your own external reference resistor, if you prefer. In two wire mode, only one analog input channel per resistance is required. In four wire mode, two analog input channels per resistance are required, which effectively divides the number of available channels on your device by two. Channels are paired consecutively: (0,1), (2,3), (4,5).... You only need to specify the first channel in the channel list. Read the DNA-STP-AIU manual to learn how to connect your resistance.

```
// Add 3 channels (0, 2 and 4) to the channel list and configure
// them to measure a resistance between 0 and 100 Ohms.
// The resistance is connected to the DAQ device using
// four wires, the excitation voltage is 5V, and the reference
// resistor is the 20kOhm resistor built-into the DNA-STP-AI-U.

MySs.CreateResistanceChannel("pdna://192.168.100.2/dev0/Ai0,2,
                             4, , 0, 1000.0,
                             UeiFourWires, 5.0,
                             UeiRefResistorBuiltIn, 20000.0,
                             UeiAIChannelInputModeDifferential);
```

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

2.2.4 RTD Measurement

RTD measurements are configured using the Session object method “CreateRTDChannel”.

RTD sensors are resistive sensors whose resistance varies with temperature. Knowing the resistance of an RTD, we can calculate the temperature using the “Callendar Van-Dusen” equations.

RTD sensors are specified using the "alpha" (a) constant. It is also known as the temperature coefficient of resistance, which defines the resistance change factor per degree of temperature change. The RTD type is used to select the proper coefficients A, B and C for the Callendar Van-Dusen equation, which is used to convert resistance measurements to temperature.

To measure the RTD resistance, we need to know the amount of current flowing through it. We can then calculate the resistance by dividing the measured voltage by the known excitation current.

To measure the excitation current, we measure the voltage from a high precision reference resistor whose resistance is known.

The reference resistor is built-into the terminal block if you are using a DNA-STP-AI-U, but you can provide your own external reference resistor, if you prefer.

In addition, you must configure the RTD type and its nominal resistance at 0° Celsius, as shown in the following example.

```
// Add 4 channels (0 to 3) to the channel list and configure
// them to measure a temperature between 0.0 and 200.0 deg. C.
// The RTD sensor is connected to the DAQ device using
// two wires, the excitation voltage is 5V, and the reference
// resistor is the 20kOhm resistor built-into the DNA-STP-AI-U.
// The RTD alpha coefficient is 0.00385, the nominal resistance at 0°
// C is 100 Ohms, and the measured temperature will be returned in
// degrees Celsius.

MySession.CreateRTDChannel(“pdna://192.168.100.2/dev0/Ai0:3”,
                           0, 1000.0,
                           UeiTwoWires,
                           5.0,
                           UeiRefResistorBuiltIn,
                           20000.0,
                           UeiRTDType3850,
                           100.0,
                           UeiTemperatureScaleCelsius,
                           UeiAIChannelInputModeDifferential);
```

2.3 Configuring the Timing

You can configure the AI-207 to run in simple mode (point by point) or buffered mode (ACB mode).

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

In buffered mode, the delay between samples is determined by the AI-207 on-board clock.

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

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

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

2.4 Reading Data

Reading data from the AI-207 is done using a reader object. There is a reader object to read raw data coming straight from the A/D converter. There is also a reader object to read data already scaled to volts or mV/V.

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

```
// Create a reader and link it to the session's stream
CueiAnalogScaledReader reader(session.GetDataStream());
// read one scan, the buffer must be big enough to contain
// one value per channel
double data[2];
reader.ReadSingleScan(data);
```

2.5 Cleaning-up the Session

The session object cleans itself up when it goes out of scope or when it is destroyed. However, you can also clean up the session manually (to reuse the object with a different set of channels or parameters), as follows.

```
session.CleanUp();
```

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

Chapter 3 Programming with the Low-Level API

The low-level API offers direct access to PowerDNA DAQBios protocol and allows you to directly access device registers.

We recommend that you use the UeiDaq Framework (*see Chapter 2*), which is easier to use.

You should need to use the low-level API only if you are using an operating system other than Windows.

Please refer to the `API Reference Manual` document under:

Start » Programs » UEI » PowerDNA » Documentation

for pre-defined types, error codes, and functions for use with this layer.

3.1 Configuration Settings

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

Not all configuration bits apply to the AI-207 layer.

The following bits are used:

```
#define DQ_LN_IRQEN      (1L<<10)    // enable layer irqs
#define DQ_LN_PTRIGEDGE1 (1L<<9)     // stop trigger edge MSB
#define DQ_LN_PTRIGEDGE0 (1L<<8)     // stop trigger edge: 00 -
software,
                                     // 01 - rising, 02 - falling
#define DQ_LN_STRIGEDGE1 (1L<<7)     // start trigger edge MSB
#define DQ_LN_STRIGEDGE0 (1L<<6)     // start trigger edge: 00 -
software, 01 - rising,
                                     // 02 - falling
#define DQ_LN_CLCKSRC1   (1L<<3)     // CL clock source MSB
#define DQ_LN_CLCKSRC0   (1L<<2)    // CL clock source 01 - SW, 10 -
HW, 11 -EXT
#define DQ_LN_ACTIVE     (1L<<1)     // "STS" LED status
#define DQ_LN_ENABLED    (1L<<0)     // enable operations
```

For streaming operations with hardware clocking, the user has to select the following flags:

```
DQ_LN_ENABLE | DQ_LN_CLCKSRC0 | DQ_LN_STREAMING | DQ_LN_IRQEN |
DQ_LN_ACTIVE
```

DQ_LN_ENABLE enables all layer operations.

DQ_LN_CLCKSRC0 selects the internal channel list clock (CL) source as a time base. The AI-207 layer supports the CL clock only where the time between consecutive channel readings is calculated by the rule of maximizing setup time per channel. If you'd like to select the CL clock from an external clock source such as the SYNCx line, set DQ_LN_CLCKSRC1 as well.

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

```
Aggregate rate = Per-channel rate * Number of channels
```

3.2 Channel List Settings

The AI-207 layer has a very simple channel list structure, as shown below:

Bit	Name	Purpose	Macro
31	DQ_LNCL_NEXT	Tells firmware that there is a “Next Entry” in the channel list	
20	DQ_LNCL_TSRQ	Request timestamp as the next data point	
11..8		Gain	DQ_LNCL_GAIN()
7..0		Channel number	

Gains are different for different options of the AI-207 layer, as listed in the following table.

Layer type	Range	Gain	Gain Number	Min. Allowed Settling Time, us
AI-207	±10V	1	0	60
	±5V	2	1	70
	±2.5V	4	2	80
	±1.25V	8	3	90
	±1V	10	4	100
	±500mV	20	5	120
	±250mV	40	6	140
	±125mV	80	7	160
	±100mV	100	8	180
	±50mV	200	9	200
	±25mV	400	10	240
	±12.5mV	800	11	280

The Minimum Allowed Settling Time is the shortest time for which the firmware allows a channel to settle. When the scan rate and channel are programmed, the firmware allocates the minimum time for each channel depending on the gain selected, and then stretches the settling time as much as possible to utilize at least 2/3 of the time between scan clocks.

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

Appendices

A. Appendix A - Accessories

The following cables and STP boards are available for the AI-207 layer.

DNA-CBL-37

3ft, 37-way flat ribbon cable; connects DNA-AI-207 to DNA-STP-37

DNA-CBL-37S

3 ft, 37-way round, shielded cable

DNA-STP-37

37-way screw terminal panel; requires DNA-CBL-37

DNA-STP-AI-U

Universal screw-terminal panel with embedded CJC.

DNA-STP-AI-207TC

Screw terminal panel for use with the DNA-AI-207 and thermocouples. The panel provides open thermocouple detection as well as the cold-junction compensation measurement.

B. Appendix B - Layer EEPROM structure

Layer configuration is stored in the EEPROM. Use the `DqCmdSetParam()/ DqCmdGetParam()` functions to access the EEPROM.

The AI-208 layer EEPROM contains standard layer configuration as well as calibration and channel naming data.

```
typedef struct {
    DQEECMNDEVS ee;
// standard layer data
    DQCALSET_208_ calset;
// calibration data
    DQOPMODEPRM_208_ opmodeprm;
// operation mode settings
    DQCNAMES_208_ cname;
// channel names
} DEVEEPROM_208_, *pDEVEEPROM_208_;
```

Channel names length can be up to twenty characters in length and are stored in the DQCNAMES_208_ structure.

```
#define DQ_AI208_NAMELEN   20  // maximum length of the channel name
                                  // (trailing 0 isn't included)

/* channel names */
typedef struct {
    char cname[DQ_AI208_CHAN][DQ_AI208_NAMELEN];
```

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

```
} DQCNAMES_208_, *pDQCNAMES_208_;
```

Calibration values are stored in the following structure:

```
/* specific device structure - calibration values
*/
typedef struct {
    uint8 cal[DQ_AI208_CALDACS];  /* four
calibration DAQs */
    uint16 caloffs[DQ_AI208_GAINS];/* offsets for
every gain */
} DQCALSET_208_, *pDQCALSET_208_;
```

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.

Index

© Copyright 2010 all rights reserved
United Electronic Industries, Inc.