United Electronic Industries

The High-Performance Alternative

DNx-DIO-432/433

—

User Manual

32-channel, Current Sink (432)/Source (433)
Digital Output Layers
for PowerDNA Cube and PowerDNR RACKtangle Systems

Release 4.6

June 2013

PN Man-DNx-DIO-432/433-613

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.

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 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:

Support:	support@ueidaq.com
Web-Site:	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 document outlines the feature-set of the DNR- and DNA-DIO-432/433 layer and its use for synchronous serial-line communications applications.

1.1 Organization of Manual

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

- **Introduction**
  This section provides an overview of the DIO-432 and DIO-433 interface board features, device architecture, and connectivity.
- **Programming with the High-Level API**
  This chapter provides an overview of the how to create a session, configure the session, and format relevant data with the Framework API.
- **Programming with the Low-Level API**
  Describes low-level API commands for configuring and using the DIO-432/433 series layer for serial operating modes.
- **Appendix A - Accessories**
  This appendix provides a list of accessories available for use with the DNx-DIO-432/433 serial-line communication interface board.
- **Appendix B - Soft Start/Stop/PWM**
  This appendix describes the Soft Start/Soft Stop/Constant PWM feature for gradually increasing/decreasing and controlling average power applied to an output load
- **Appendix C - Sync Interface**
  This appendix describes the use of the PowerDNx Sync Interface to supply a start/stop trigger to any layer in your system.
- **Index**
  This is an alphabetical listing of the topics covered in this manual.

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**.”

Text formatted in `fixed` typeface generally represents source code or other text that should be entered verbadim 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^{TM} 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 The DIO-432/433 Interface Board

The DNx-DIO-432 and -433 32-channel digital output layers are designed for a wide variety of digital control applications. Each channel in the DIO-432 uses low-side FETs in the output circuit shown in **Figure 1-1** to switch voltages in the range of 3.3 to 36VDC.The layer requires an external 3.3 to 36VDC power supply for the load. The DIO-433 uses high-side FETs in the circuit shown in **Figure 1-2** and also requires an external power supply.

Each channel is rated for continuous operation at 600 mA with an output voltage drop of less than 550 mV. The 32 DIO-432 outputs are configured as current sink outputs; the 32 DIO-433 outputs are configured as current source outputs.

No separate clock or trigger input is provided, but the board may be controlled through the SYNC interface. Refer to “Appendix C” on page 20 for a detailed description of the SYNC interface.

Both layers will not be damaged by occasional current spikes up to 2A/channel, but overall sink (432)/source(433) current should be limited to 24A per layer. Ensure that the power supply can supply enough current for the connected load.

Current sense resistors mounted in the output circuits for each channel enable the board to monitor both current and voltage continuously and to detect and flag short circuits, open circuits, and other “off-normal” operating conditions for every output circuit. The monitoring feature notifies the operator and/or the host CPU via an interrupt whenever a fault occurs. This capability is a powerful diagnostic tool that enables a technician to quickly detect a fault condition and take immediate corrective action. Refer to the Framework API Reference Manual for a more detailed description of how these features can be applied.

The output current monitor also provides over- and under-current protection. This feature lets you select the trip current level and also the duration of a permitted overload, which may be set as short as 10 milliseconds. When a set point is exceeded, the circuit is shut down. Each board provides 350 Vrms isolation between I/O and Cube and any other installed layers.

The layers offer update rates up to 1 kHz and simplify software writes by transferring the states of all outputs as a single, 32-bit word. The DIO-432 and the -433 are fully supported by the UEIDAQ Framework API, which provides a simple and complete software interface to all popular programming languages, operating systems, and data acquisition/control application packages such as LabVIEW, DASYLab, and MATLAB.

Any channel not used as a digital output may be used as a general purpose analog voltage input.
For the DIO-432 (Low-Side FET), connect the voltage input signal between DOut LO (up to +40 VDC) and DGND (0 VDC), as shown in **Figure 1-1**. The signal level should be -0.4 to +40VDC input range relative to DGND.
For the DIO-433 (High-Side FET), connect the voltage signal between DOut HI (up to -40 VDC) and GND (0 VDC), as shown in **Figure 1-2**. (Note that the voltage input signal range is +.4 to -40VDC relative to Vcc.

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

The schematic for the low-side FET digital outputs of the 432 is shown in **Figure 1-1**.

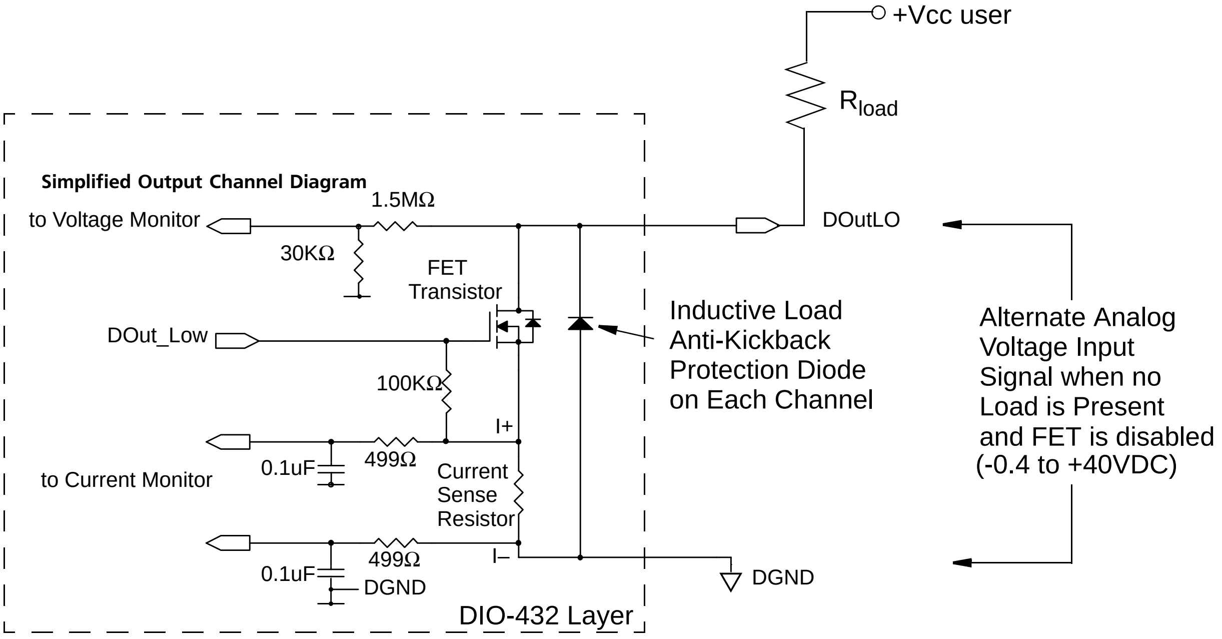


Figure 1-1. Schematic for Low-Side FET Digital Output in DIO-432

As shown in **Figure 1-1** and **Figure 1-2**, the DIO-432 reports a positive voltage measurement between DGND and DOUT pins. The DIO-433 reports negative voltage values measured between Vcc and DOUT pins. Both layers return current (measured at the I+ and I– pins of the sense resistor) as a positive number when the FET is open.

The schematic for the high-side FET digital outputs of the 433 is shown in **Figure 1-2**.

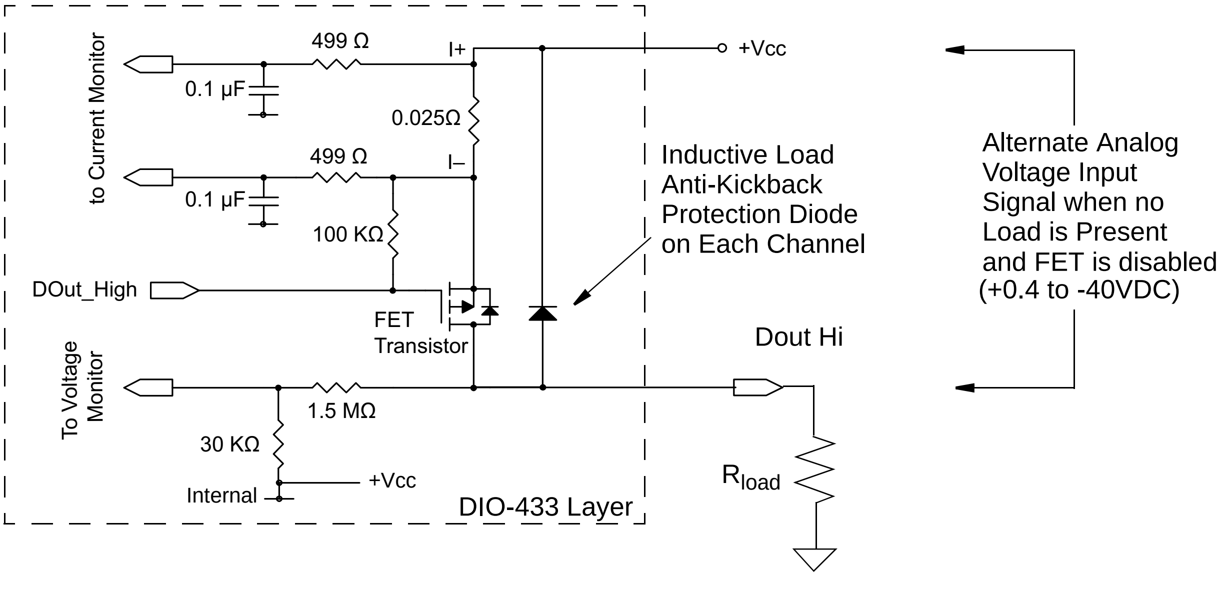


Figure 1-2. Schematic for High-Side FET Digital Output in DIO-433

1.3 Features

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

- 32 digital output channels:
  - 32 high-side channels (DNA-DIO-433 and DNR-DIO-433)
  - 32 low-side channels (DNA-DIO-432 and DNR-DIO-432)
- Current sink outputs (DNx-DIO-432)
- Current source outputs (DNx-DIO-433)
- Overcurrent Protection user-programmable from 50mA to 2A and stored in EEPROM (lower on DIO-432/433-800 boards)
- Overcurrent Response Time user-programmable from 10 to 5000 ms
- Output Throughput Rate of 1000 samples per second (max)
- 600 mA per channel of continuous output current, 2A momentary
- Maximum total current 24A aggregate for all channels (per layer)
- Built-in Monitoring of voltage and current on each output channel provides fault detection and simplifies system diagnostics (1% or better accuracy at ±50V input range)
- Wide 3.3 V to 36 V Operating Range (requires external power supply for load)
- Inductive Load Kickback Protection Diodes on every channel
- Intelligent Electronic Current Monitoring (±0.5% of F.S.accuracy) on every channel with circuit breaking via FET disable
- SYNC interface support — no clock/trigger input, but may be clocked/ triggered via SYNC lines
- Power Consumption of the layer is 2.5W with all outputs driving 600 mA
- Overcurrent and undercurrent per channel limits
- Interrupt on over- or undercurrent condition
- Auto-enable option is selectable per channel. It will try to restore disabled channel functionality after a programmable delay that can be up to 65 seconds long (settable in 15 nS increments)
- Guaranteed output disabled (OFF) state under the following conditions: initial power-up, external power OFF, overload detected
- PWM (slow-ON/slow-OFF or constant PWM) function on every output
- UEI Framework Software API may be used with all popular Windows programming languages and most real time operating systems such as RT Linux, RTX, or QNX and graphical applications such as LabVIEW, MATLAB, DASYLab and any application supporting ActiveX or OPC.

1.4 Indicators

A photo of the DIO-432/433 unit is illustrated below.

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.

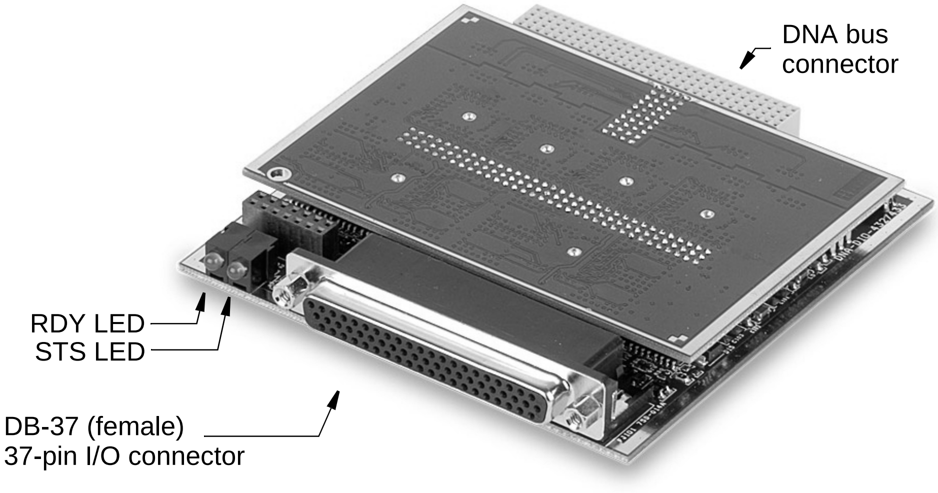


Figure 1-3. The DNA-DIO-432/433 Layer

1.5 Specification

The technical specification for the DIO-432/433 is provided in the table below:

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

Number of channels	32 digital outputs
Output configuration	Current sink
Output port configuration	Single 32-bit word
Output Drive	600 mA per channel continuous; 3.5 A peak (100 mS max)
Output ON voltage	< 550 mV @ 600 mA (incl. std 3' cable)
Output ON impedance	< 0.9 Ohm (including std 3' cable)
Output OFF impedance	> 1 Meg Ohm
Output OFF leakage	< 25 µA
Overvoltage protection	±40 VDC (reverse current must be limited to 1 A to prevent damage)
Overcurrent protection Current Limit Overload response time	 50 mA - 2 A 10-5000 ms (user programmable)
Output Monitoring Configuration Voltage Accuracy Current Accuracy	 Multiplexed ±10 mV max (sampled at 2 Hz) ±1 mA, max (sampled at 10 Hz)
Soft-Start/Stop duration	256 µS to 5 seconds
Steady State PWM output	0 to 100% in 0.4% increments. (Minimum period is 256 µSec)
Output Throughput Rate	1 kHz max
Power up / reboot state	Off
Power dissipation	< 2 W, not including output switches
Isolation	350 Vrms
Operating Temp. Range	Tested -40 to +85 °C
Operating Humidity	95%, non-condensing
Vibration *IEC 60068-2-6* *IEC 60068-2-64*	5 g, 10-500 Hz, sinusoidal 5 g (rms), 10-500 Hz, broad-band random
Shock *IEC 60068-2-27*	50 g, 3 ms half sine, 18 shocks @ 6 orientations 30 g, 11 ms half sine, 18 shocks @ 6 orientations
MTBF	260,000 hours

The technical specifications for the DNx-DIO-433 layer are the same as those for the -432 except for the following items:

Specificaton Item	**432**	**433**
Output Configuration	Current Sink	Current Source
Output Drive	600 mA per channel continuous, 3.4A peak	600 mA per channel continuous, 2.7A peak (10% duty cycle) and 100ms duration max.
Output ON Voltage Drop	<550 mV at rated 600 mA	<500 mV at rated 600 mA
Output ON Impedance	<0.9 Ohm	<1.0 Ohm

1.6 Device Architecture

A block diagram of a DNx-DIO-432 or -433 layer is shown below.

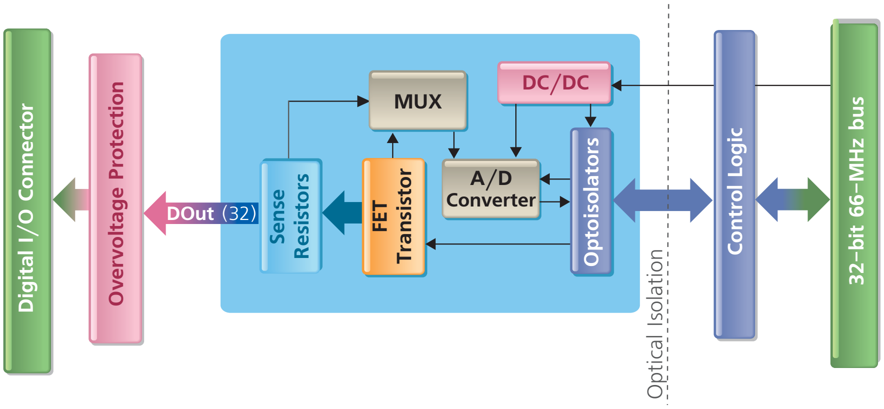


Figure 1-4. DNA/DNR-DIO-432/433 Logic Block Diagram

As shown in **Figure 1-4**, layer logic is divided into isolated and non-isolated sections. The non-isolated logic complies with the full Control Logic Interface standard. The isolated side handles all functions associated with switching the output loads, sensing current and voltage, and providing over- and under-current and voltage monitoring and control. It also provides anti-kickback protection for inductive loads.

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

Current monitoring is accomplished with a 24-bit ADC that uses 0.1V as a reference voltage, increasing the dynamic range of the device. Currents up to a maximum of ±2A can be detected by the layer. If the DC/DC module is disabled, the FETs are disabled and the load is disconnected.

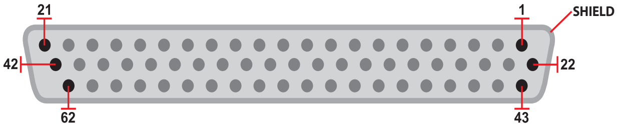
The DIO-432 and -433 are both two-board units, comprised of a layer-specific base board (made in both DNA and DNR versions) and a plug-in layer-specific ADC board. The base boards for 432 and 433 layers use some different components, but the major difference between them is that the DIO-432 base and the DIO-433 base are fabricated in two versions, one designed for installation in a PowerDNA Cube (DNA version) and another (DNR version) designed for insertion into a RACKtangle rack enclosure. The DNA versions have a 120-pin connector for the Vertical Bus of a Cube and the DNR versions have a connector that plugs into a backplane in the rack enclosure. The DNA versions also have a jumper block for selecting a layer position within a Cube. The DNR versions do not need this feature.

The ADC boards are also fabricated in two versions, one for both DNA and DNR DIO-432 and one for both DNA and DNR DIO-433 layers.

The base boards contains the FETs, sensing resistors, DC/DC module, and CLI logic; the plug-in boards contain the ADCs, buffers, and associated circuits.

1.7 Wiring & Connectors

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



DNx-DIO-432 Pinout

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

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

DNx-DIO-433 Pinout

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

Note:

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

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

Figure 1-5. DNx-DIO-432/433 Pinout Diagram

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

1.8 Setting Operating Parameters

For detailed instructions for configuring the layer and setting operating modes and parameters, refer to the Framework Functions **DqAdv432SetAll** and **DqAdv432SetAll**, which are described in the Framework API Reference Manual.

1.9 Configuring the Circuit Breaker Function

Referring to the circuits of **Figure 1-1** and **Figure 1-2**, the voltages from current sensing resistors are fed to 24-bit sigma-delta ADCs. The ADC outputs are then processed in the logic to perform a virtual circuit breaker function. The outputs are first compared to preset limits. If they exceed the limits, the FETs are turned OFF and the output circuit is shut down. Depending on how the channel is configured, the shutdown may either be immediate or delayed by a programmable time or by a cumulative number of detected faults.

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

1.10 Configuring ADC Conversion Speed

The speed and resolution of the ADC are user-programmable in the range from 0.6 to 293 Hz. Refer to the description of the function **DqAdv432SetAll** in the Framework API Reference Manual.

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

Table 1-2 provides the ADC speeds, the time required to react to over-current and shut down the output circuit using the circuit breakers, and the accuracy. If we plot this data, we can see that accuracy is best at low sampling rates, and that the reaction time is quickest at high sampling rates.

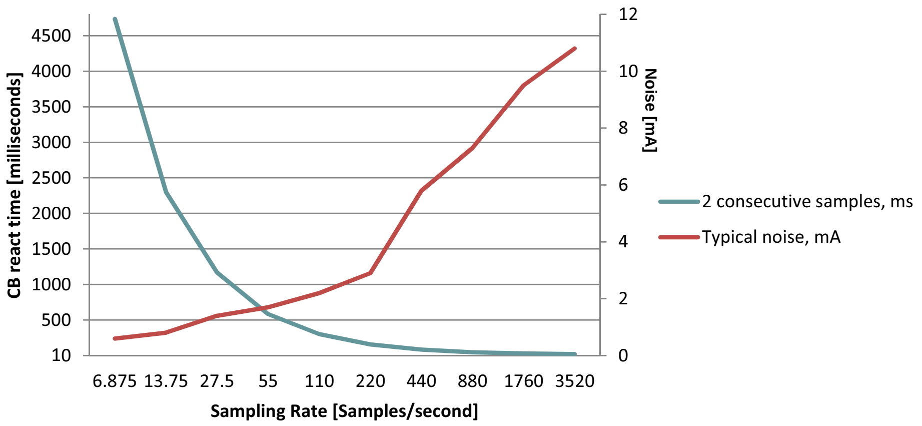


Figure 1-6. Sampling rate, CB react time, and noise

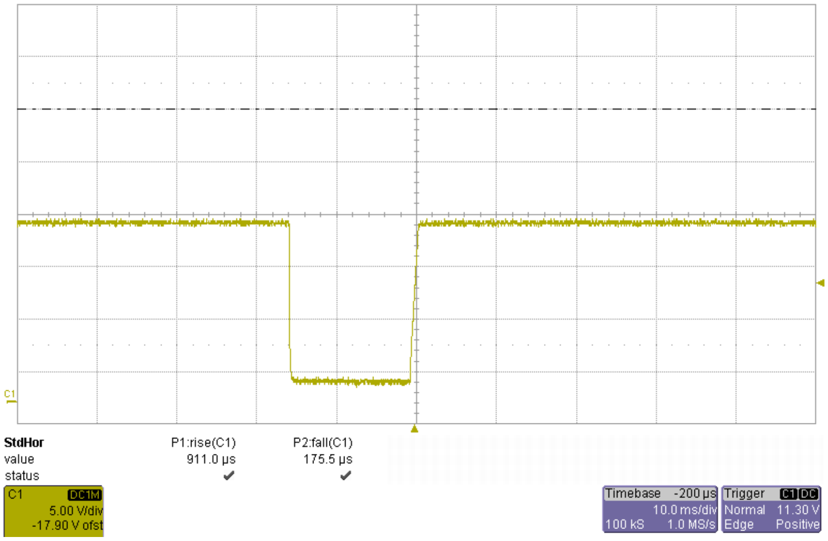
As a guideline, you should choose a sampling rate that is low enough to meet your accuracy need, but fast enough to allow the circuit breaker the time needed to disconnect in case of over-current. Note that the over-current limit is software configurable with the **DqAdv432SetLimit** function. Set sampling rate and limits to avoid over-current for a prolonged period; failing to do so may permanently damage your external equipment and/or the layer.

The table below provides sample rate, typical circuit breaker disconnect time in milliseconds, and noise in milliamps.

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

Table 1-2. Sample rate, disconnect time, and accuracy

By default, the DNx-DIO-432 is configured to sample at 3520Hz and trips the circuit breaker at 2A in less than 20ms, as shown below:



StdHor	P1:rise(C1)	P2:fall(C1)
value	911.0 µs	175.5 µs
status	✓	✓

C1 DC1M
5.00 V/div
-17.90 V ofst

Timebase -200 µs
10.0 ms/div
100 kS 1.0 MS/s

Trigger C1 DC
Normal 11.30 V
Edge Positive

Figure 1-7. DIO-432 disconnect time (default setting)

Chapter 2 Programming with the High Level API

This section describes how to control the DNx-DIO-432/433 using the UeiDaq Framework High Level 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 or LabVIEW.

The following section focuses on the C++ API, but the concept is the same no matter what programming language you use.

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

2.1 Creating a Session

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

```
// create a session object

CUeiSession session;
```

2.2 Configuring the Resource String

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

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

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

For example, the following resource string selects digital output lines 0,1,2,3 on device 1 at IP address 192.168.100.2: “pdna://192.168.100.2/Dev1/Do0:3”

2.3 Configuring for Digital I/O

The DIO-432/433 can be configured for digital output.

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

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

The DIO-432/433 is known as an intelligent digital output device. It can monitor the current flowing through each of its digital lines and open or close a line when the current goes above or below specified current limits.

You can configure the device so that when an over or under current condition occurs, it will attempt to close the connection after a programmed delay.

The device is also capable of generating a pulse train for each rising and/or falling edge, thus allowing for soft start and/or stop.

The following call configures the digital output port of a DNx-DIO-432/433 set as device 1:

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

It configures the following parameters:

- **Under-current limit:** when the current goes below this limit, the line opens.
- **Over-current limit:** when the current goes above this limit, the line opens.
- **Current sampling rate:** the rate at which the DNx-DIO-432/433 monitors current. This rate has a direct influence on how fast the DNx-DIO-432/433 reacts to an under or over-current condition.
- **The retry status:** specifies whether the DNx-DIO-432/433 attempts to close the circuit after an over or under current condition.
- **The retry rate:** specifies how often the DNx-DIO-432/433 attempts to close the circuit.

The pulse width modulation features are programmable on a per-output line basis. There are a few methods available in the CUeiDOProtectedChannel class to program the behavior of some of the output lines.

First, get a pointer to the channel object:

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

All lines will run at the same frequency. The following code programs the pulse period to 50 us:

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

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

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

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

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

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

Also specify the length in us of the pulse train generated on lines configured for soft start/stop mode::

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

NOTE: When a line is configured to generate pulses continously, write operations (as described in Section 2.5 below) are ignored. The line will keep generating pulses no matter what value is written to it.

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

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

2.4 Configuring the Timing

You can configure the DIO-432/433 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 DNx-DIO-432/433 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 for point-by-point (simple mode)
session.ConfigureTimingForSimpleIO();
```

2.5 Writing Data

Writing data is done using a writer object.

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

```
// create a writer and link it to the session’s stream
CUeiDigitalWriter writer(session.GetDataStream());
// to write a value, the buffer must contain one value per channel
uInt32 data = 0xFEFE;
// write one scan, the buffer must contain one value per channel
writer.WriteSingleScan(&data);
```

2.6 Monitoring the Current

You can monitor the current measured at each digital line.

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

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

```
// create & configure session

CUeiSession aiSs;
aiSs.CreateAIChannel("pdna://192.168.100.2/Dev1/Ai0:3"
                      -10.0, 10.0,
                      UeiAIChannelInputModeDifferential);
aiSs.ConfigureTimingForSimpleIO();

// create & configure reader; create buffer

CUeiAnalogScaledReader aiReader(aiSs.GetDataStream());
double currents[8];

// read monitored current into buffer

aiReader.ReadSingleScan(currents);
```

2.7 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 illustrates how to program the PowerDNA cube using the low-level API. The low-level API offers direct access to PowerDNA DAQBios protocol and also allows you to access device registers directly.

However, we recommend that, when possible, you use the UeiDaq Framework High-Level API, because it is easier to use. You should need to use the low-level API only if you are using an operating system other than Windows.

For additional information about low-level programming of the DIO-432/433, please refer to the PowerDNA API Reference Manual document under:

Start » Programs » UEI » PowerDNA » Documentation

Refer to the PowerDNA API Reference Manual on how to use the following low-level functions of DIO-432/433, as well as others related to cube operation:

Function	Description
`DqAdv432GetAll`	Gets two structures with current values and calibration data.
`DqAdv432SetAll`	Sets most of the configuration parameters.
`DqAdv432SetLimit`	Sets the over- and under-current limits for a channel.
`DqAdv432SetPWM`	Configures the PWM mode for a channel.

Appendix A

A. Accessories

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

DNA-CBL-62

This is a 62-conductor round shielded cable with 62-pin male D-sub connectors on both ends. It is made with round, heavy-shielded cable; 2.5 ft (75 cm) long, weight of 9.49 ounces or 269 grams; up to 10ft (305cm) and 20ft (610cm).

DNA-STP-62

The STP-62 is a Screw Terminal Panel with three 20-position terminal blocks (JT1, JT2, and JT3) plus one 3-position terminal block (J2). The dimensions of the STP-62 board are 4w x 3.8d x1.2h inch or 10.2 x 9.7 x 3 cm (with standoffs). The weight of the STP-62 board is 3.89 ounces or 110 grams.

DB-62 (female)
62-pin connector:

62	to J2	**42**	to J2	**21**	to J2
61	to JT1	**41**	to JT1	**20**	to JT1
60	to JT1	**40**	to JT1	**19**	to JT1
59	to JT1	**39**	to JT1	**18**	to JT1
58	to JT1	**38**	to JT1	**17**	to JT1
57	to JT1	**37**	to JT1	**16**	to JT1
56	to JT1	**36**	to JT1	**15**	to JT1
55	to JT2	**35**	to JT1	**14**	to JT1
54	to JT2	**34**	to JT2	**13**	to JT2
53	to JT2	**33**	to JT2	**12**	to JT2
52	to JT2	**32**	to JT2	**11**	to JT2
51	to JT2	**31**	to JT2	**10**	to JT2
50	to JT2	**30**	to JT2	**9**	to JT2
49	to JT2	**29**	to JT2	**8**	to JT2
48	to JT3	**28**	to JT3	**7**	to JT2
47	to JT3	**27**	to JT3	**6**	to JT3
46	to JT3	**26**	to JT3	**5**	to JT3
45	to JT3	**25**	to JT3	**4**	to JT3
44	to JT3	**24**	to JT3	**3**	to JT3
43	to JT3	**23**	to JT3	**2**	to JT3
		22	to JT3	**1**	to JT3

JT3 — 20-position **terminal block:**	**JT2** — 20-position **terminal block:**	**JT1** — 20-position **terminal block:**	**J2** — 5-position **terminal block:**	
○ 22	○ 55	○ 20	**5**	SHIELD
○ 1	○ 13	○ 41	**4**	GND
○ 43	○ 34	○ 61	**3**	62
○ 23	○ 54	○ 19	**2**	42
○ 2	○ 12	○ 40	**1**	21
○ 44	○ 33	○ 60		
○ 24	○ 53	○ 18		
○ 3	○ 11	○ 39		
○ 45	○ 32	○ 59		
○ 25	○ 52	○ 17		
○ 4	○ 10	○ 38		
○ 46	○ 31	○ 58		
○ 26	○ 51	○ 16		
○ 5	○ 9	○ 37		
○ 47	○ 30	○ 57		
○ 27	○ 50	○ 15		
○ 6	○ 8	○ 36		
○ 48	○ 29	○ 56		
○ 28	○ 49	○ 14		
○ **GND**	○ 7	○ 35		

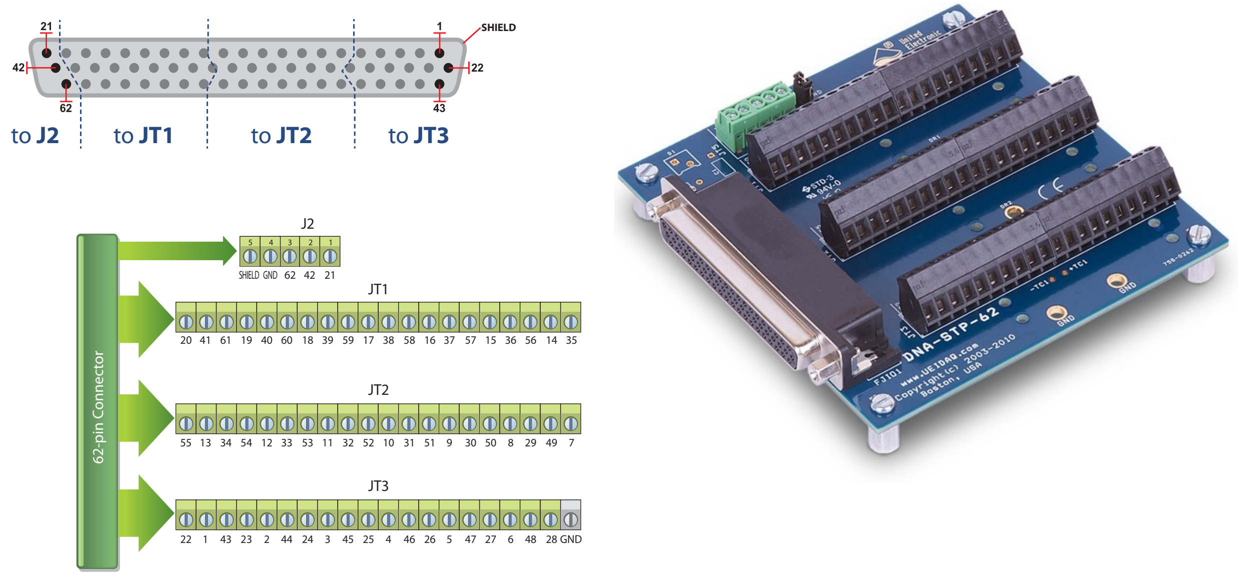


Figure A-1. Pinout and photo of DNA-STP-62 screw terminal panel

Appendix B

Soft Start/Soft Stop/Constant PWM Feature Description

B.1 Soft Start

The DNx-DIO-432 and 433 layers offer a per-channel configured PWM start/stop mode. The graphic shown in **Figure A-1** shows a trace of a typical soft-start of the output (rising edge) done in 16 PWM period steps with the period set to approximately 6mS. This feature is useful in preventing premature burnout of devices such as incandescent bulbs caused by too rapid heating on startup.

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

For detailed instructions on setting parameters for normal/soft start/soft stop/constant PWM operation, refer to the function **DqAdv432SetPWM** in the Framework API Reference Manual.

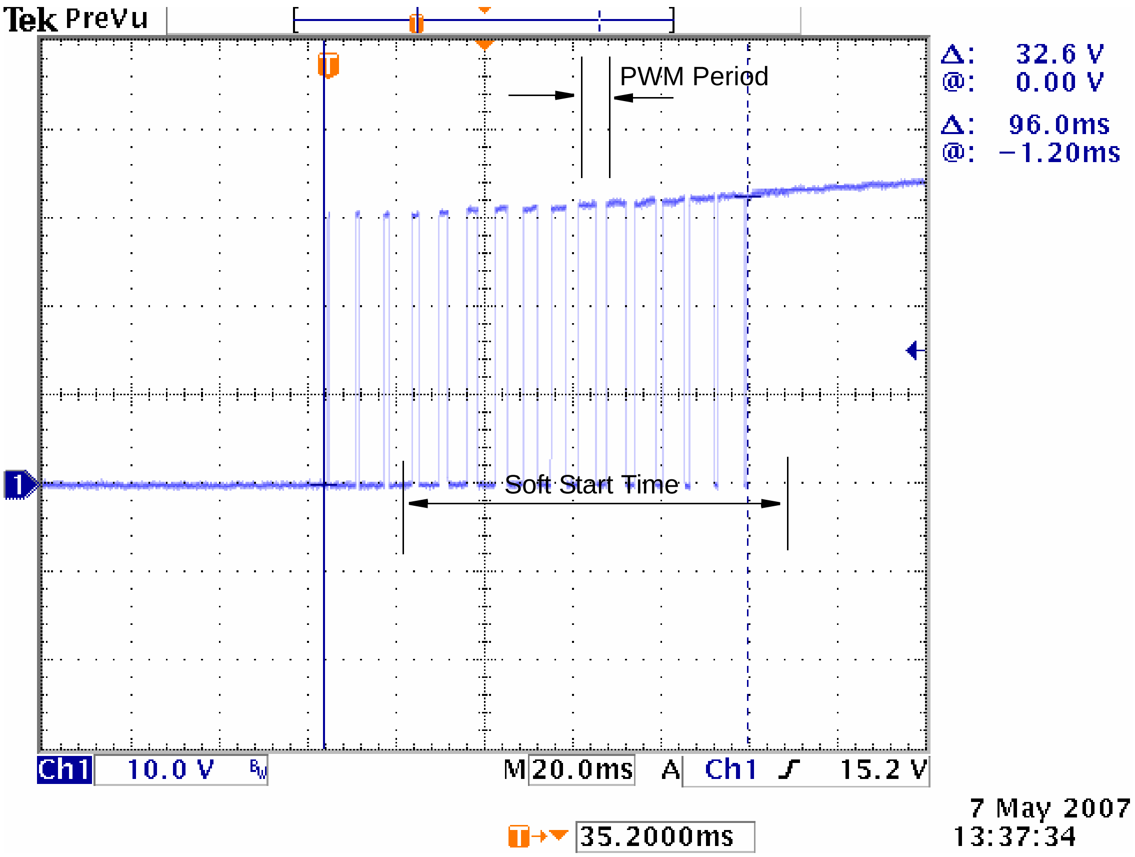


Figure B-1. Typical Soft Start Cycle

B.2 Soft Stop

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

B.3 PWM Mode

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

B.4 Example – Soft Start Code

The code below is excerpted from Sample 432_433.c found in the DAQLib_Samples folder:

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

#define IOM_IPADDR0          "192.168.100.2"
#define DEVN                 0
#define CHANNELS             DQ_DIO432_CHAN
#define OVERCURRENTLIMIT   (1.000) // over  current limit (0...2A)
#define UNDERCURRENTLIMIT  (-0.01) // under current limit (-2A...0)
#define PWM_MODE     1 // 1-enable/0-disable PWM mode (Optional mode)
#define PWM_MODE_DUTY_CYCLE 8).    // Valid Duty cycle is set
//   with 8-bit accuracy when PWM_MODE is enabled. This value
//   sets the PWM duty cycle. For details, see NOTES 2 and 3 below.
//   In soft start/soft stop mode, (DQDIO432_PWM_SOFTSTART,
//   DQDIO432_PWM_SOFTSTOP,and DQDIO432_PWM_SOFTBOTH define the
//   number of PWM periods. For example, if <duty_cycle == 16>
//   output will produce 16 PWM periods upon changing output from
//   1/16 duty cycle to 16/16 (or 100%))
```

NOTE:

1. To avoid confusion in setting up a PWM operating mode, note that the DNx-DIO-432 output circuit has a FET to ground on each output. Therefore, an ON state would be a signal LOW. Conversely, the DNx-DIO-433 has a FET to +Vcc on each output. Therefore, an ON state for the 433 would be a signal HIGH.

2. To attain 0% duty cycle with a DIO-432, set the PWM_MODE for the channel to DQDIO432_PWM_DISABLED and set the corresponding output to 0 using the **DqAdv40xWrite()** function.

   For the DIO-433, a DUTY_CYCLE_LENGTH of 0 corresponds to 255/256% (99.6%) and 255 to 0%.

3. It is not possible to directly set duty cycle to 100% on a DIO-433. To attain 100% duty cycle with a DIO-433, set the PWM_MODE for the channel to DQDIO432_PWM_DISABLED and set the corresponding output to 1 using the **DqAdv40xWrite(**) function

Appendix C

PowerDNA Sync Interface

C.1 Introduction

The PowerDNA Sync Interface provides two capabilities that are key components of many applications.

- It allows a PowerDNA Cube to be triggered by, or synchronized to, an external event or signal.
- It allows the various I/O layers/boards within a cube to be triggered by and/or synchronized to, a variety of signals within the cube or to external signals brought in directly to an I/O layer.

PowerDNA synchronization is based on two fixed-direction signal connections (Sync In and Sync Out) which are available on the CPU layer of the Cube as well as on four bidirectional sync signals (Sync0 through Sync3) provided on the primary internal data bus of the Cube and shared by all I/O layers as well as the CPU layer. A block diagram of the system is shown in **Figure C-1.**

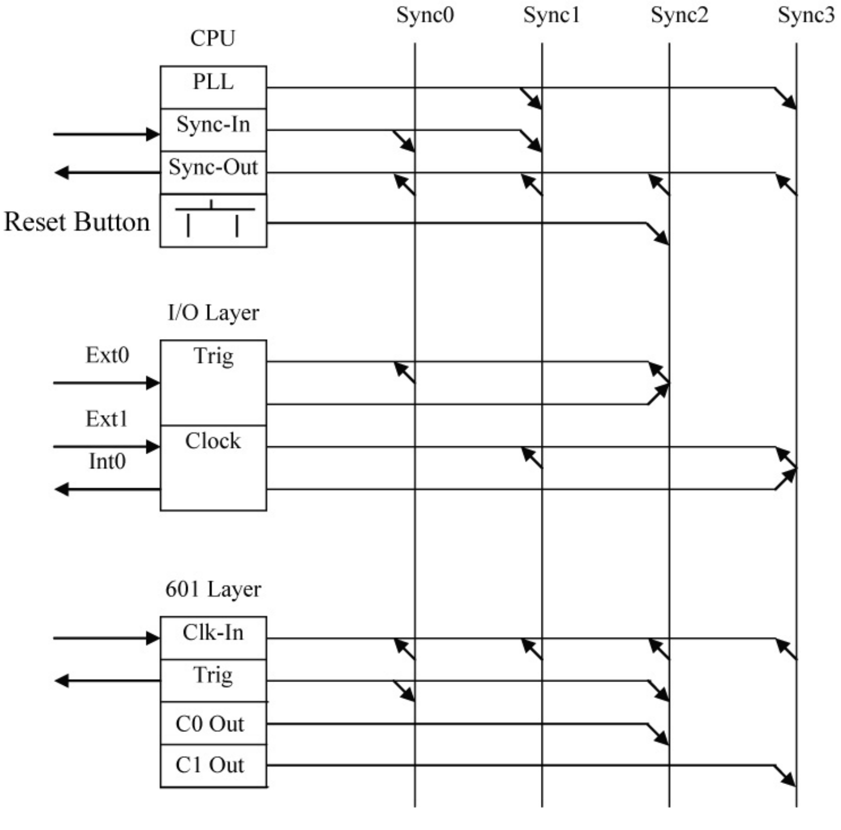


Figure C-1. Sync Interface Bus Diagram

Note that the 601 Counter/Timer layer is a unique case. The counter timer capabilities of the board make it ideal for generating various timing and synchronization signals. Therefore, it is given more extensive access to the sync bus than standard layers.

FPGA bases of the various layers, combined with the Cube firmware, can be configured to create an almost unlimited set of trigger and synchronization scenarios. Not all of these are supported in the standard released product. However, our standard trigger/synchronization model can satisfy the requirements of virtually all users.

The two-signal external Sync interface and the four-signal internal sync configuration are described in the two following sections, respectively.

C.2 External Sync/Trigger interface

Each PowerPC PowerDNA Cube provides an external Sync connector on the front panel (below the reset button, above the first I/O layer). Each DNR-12-1G module mounted in a RACKtangle rack enclosure has a sync connector on its front panel next to the reset button. These Sync interface signals may be monitored or controlled by the logic on the processor board of the Cube, or they may be connected directly to internal Sync signals shared by the internal I/O layer boards. The remainder of this section describes the external Sync interface. Please refer to Section 1.3 for details on the Cube internal sync bus.

The external Sync interface provides four connections. The Sync interface pins share a common ground, but are fully isolated from the Cube itself.

- Sync In
- Sync Out
- +5 VDC (up to 10 mA)
- Ground

Sync In is a dedicated input and may be used as a trigger source for the layer or to provide an external clock source to the cube. As a trigger, it supports the following modes.

- **Trigger Mode –** Start an application on a rising or falling edge (software selectable)
- **Trigger/Stop Mode –** Start an application on a rising (or falling) edge, Stop the application on the next rising (or falling) edge.
- **Gate High Mode –** Run the application while *Sync In* is High, Stop when *Sync In* is Low
- **Gate Low Mode –** Run the application while *Sync In* is Low, Stop when *Sync In* is High
- **Direct Layer Mode –** The *Sync In* terminal does not have a direct “Cube wide” function, but is connected directly to a *Sync* pin on one of the I/O Layers.

Sync Out is a dedicated output that may be configured to output any of the following:

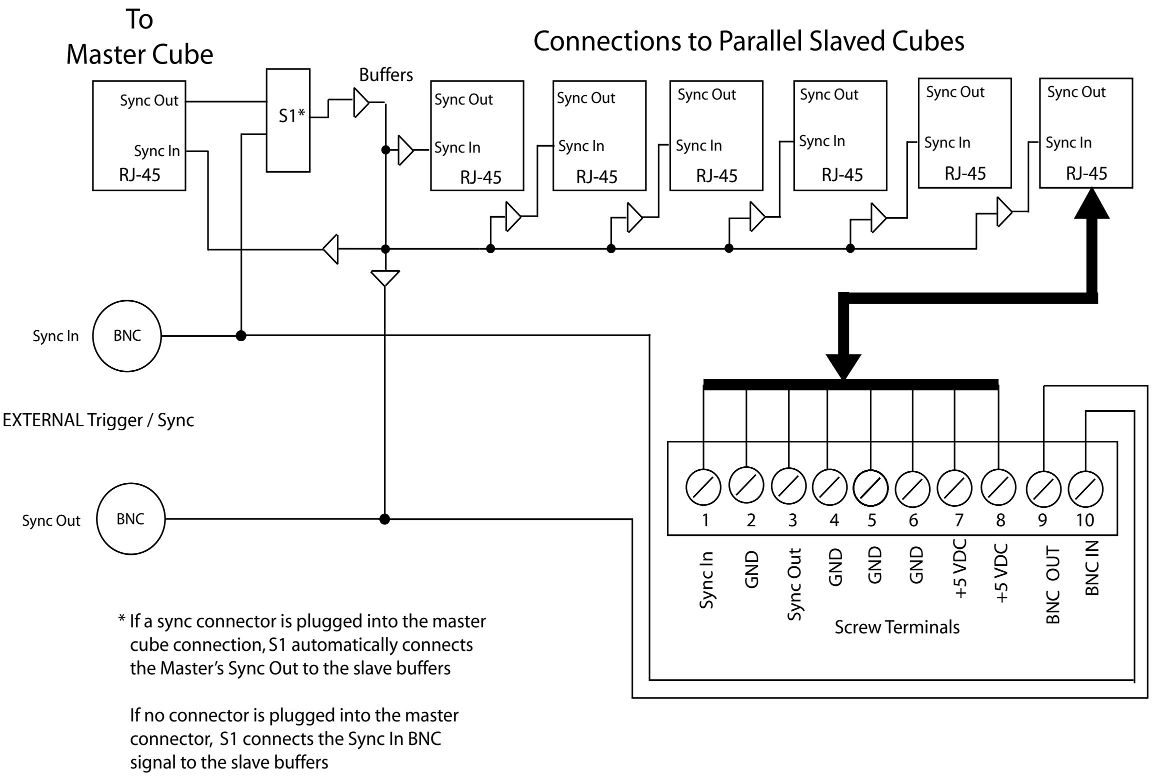
- **Sync Buffer Mode –** The Sync Out signal is simply a buffered version of Sync In
- **Ext Clock Mode –** The internal clock of the cube is brought out to the Sync out connection and may be used to synchronize clocks across cubes or throughout an application.
- **Direct Layer Mode –** The Sync Out signal is controlled by one of the I/O layers within the Cube.

There are two Sync cables available, the DNA-CBL-SYNC-RJ and the DNA-CBL-SYNC-30. The DNA-CBL-SYNC-RJ provides a Sync connector on one end and an RJ-45 connector on the other. This cable is used to connect external signals to the cube. Typically, the DNA-CBL-SYNC-RJ is plugged into the cube and also into the DNA-STP-SYNC panel. The DNA-STP-SYNC provides three sets of connections as shown below.

- The board provides a screw terminal connection for each of the four Sync signals.
- The board provides 6 parallel RJ-45 connectors. All four SYNC signals are connected in parallel as well. These parallel connections allow the user to easily connect the identical external trigger signals to multiple cubes.
- The board provides a seventh RJ-45 connector with its Sync Out pin connected to the Sync In terminals of the four parallel connectors. This will allow the Sync output of a single Cube to control other Cube Sync inputs without injecting the additive delays of multiple daisy-chained DNA-CBL-SYNC-30 connections.

A block diagram of the DNA-STP-SYNC is shown in **Figure C-2**.

The DNA-CBL-SYNC-30 cable is a 30-inch cable that simply crosses the Sync In and Sync Out connections. This connects the Sync Out of one Cube to the Sync In of the next, allowing the second cube to be slaved to the first.



* If a sync connector is plugged into the master cube connection, S1 automatically connects the Master's Sync Out to the slave buffers

If no connector is plugged into the master connector, S1 connects the Sync In BNC signal to the slave buffers

Figure C-2. DNA-STP-SYNC Block Diagram

C.3 Internal Sync Connections

Four Sync signals on the internal I/O interconnect bus are brought to each layer.

These four lines are designated as “Sync0” through “Sync3”. The diagram below shows the configuration of the four internal Sync signals and also the pinout on the I/O layers..

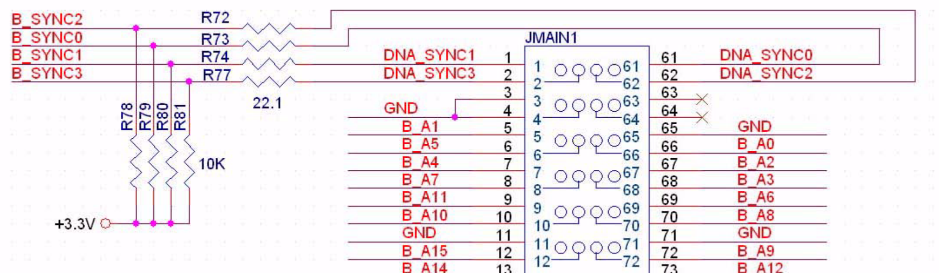


Figure C-3. Schematic of Internal Sync Connections

As you can see, each line is pulled up with 10k resistor. In the maximum PPC-8 cube, the total resistance is 1430 Ohms with a termination current of 2.3mA. These parameters prevent synchronization lines from bouncing and also ensure that proper drive is available from every layer.

The four synchronization lines have identical functionality and any of the synchronization signals can be routed to any one of the synchronization lines. These capabilities allow great flexibility of synchronization interface configuration. However, they make the synchronization model very complex.

To simplify the synchronization interface model, UEI has standardized on the following conventions.

- Sync0 – dedicated trigger input
- Sync1 – dedicated input clock or system timebase clock
- Sync2 – inter-layer triggering
- Sync3 – inter-layer clocking

This line assignment addresses virtually all anticipated synchronization requirements.

The logic on the CPU board allows either of the external Sync connections provided at the external Sync connector to be mapped to any of these four internal sync signals. In most applications, the master Sync input from the CPU board is connected either to the Sync0 or Sync1 terminals. If an I/O layer is being used as a master system trigger, however, it is expected that the external Sync Out connection would be mapped to either Sync2 or Sync3.

C.4 Layer Triggering & Clocking

A layer can be triggered using the following sources:

- Firmware executing DaqBIOS Start command
- EXT0 line
- Sync0 line
- Sync2 line

A layer can be clocked using the following sources:

- Internally
- EXT1 line (or EXT0 in a single-line layer)
- Sync1 line
- Sync3 line

A layer can feed its trigger signal to Sync2 line. A layer can feed its clock signal to the Sync3 line.

The Sync-Out line on the CPU layer can either output Sync[0..3] line or be used for alarm notification.

C.5 Use Application Cases

Use Case 1 – Starting/Stopping Multiple Layers at the Same Time (No External Trigger)

This is a very common operation in ACB and DMap modes, in which the software issues a command to begin or end an application, but multiple layers within the cube need to be synchronized. If the layers involved are in software triggering mode, the firmware proceeds as follows:

- The start sequence for all layers involved is stored (performed by prog_...() functions in the device driver)
- All I/O layer timestamp counters are reset and synchronized with the timestamp counter on the CPU layer (this is required to align data relative to timestamps)
- The start sequence is executed (normally it is a single write to LCR register of each layer involved)

Layers can be clocked internally or externally in this case.

Use Case 2 – External Trigger (via Sync Connector)

An external trigger drives Sync-In. Sync-In is sampled by the CPU layer, which then drives the internal Sync0 line. Installed I/O layers use the Sync0 line as a trigger.

Use Case 3 – External Trigger (through an I/O Layer)

An I/O layer can be used to trigger one or more of the other layers in the cube. This trigger may be based directly upon an external trigger, or based upon its own trigger or terminal count. In this case, the master layer (which provides the sync signal) drives the Sync2 line. Other layers are triggered by this Sync2 signal. Clock configuration defines what signal (software, internal, external, sync bus) will be used as a layer clock.[1]

Use Case 4 – External Clock

An external clock can be either fed into the Sync-In input on the CPU layer or the CPU-layer PLL output may be routed to SYNC-Out and then back to SYNC-In. The clock configuration defines what signal (software, internal, external, sync bus) will be used as a layer clock.

Use Case 5 – Master-Slave Clocking

In this case, one layer produces a clock signal and places it on the Sync3 line. Other layer(s) in the cube then use it as their clock. The clock source can be a standard (analog/digital) layer as well as a counter-timer (CT) layer.

Use Case 6 – Synchronous Buffered Input and Output

In this case, an analog output layer feeds its clock to the Sync3 line.

An analog input layer then uses this clock signal to synchronize its A/D sampling to the D/A layer's clock. This allows the PowerDNA to be used in stimulus/ response applications.

Use Case 7 – Sequenced Acquisition (based on the DNA-CT-601 Counter/ Timer Layer)

Sequenced acquisition can be accomplished by feeding the Sync-In trigger or clock signal into one of the counter-timers on CT-601 layers (via Sync0 and Sync1 lines) and then back out to other I/O layers (using Sync2 and Sync3 lines). This allows the PowerDNA Cube to acquire data, or output waveforms, based on a predefined sequence in the CT-601 FIFO.

[1] Note that this changes the definition of DQ_LN_CxCKSRCx bits. The bit combinations used are: 0 - software, 1 - internal, 2 - external, and 3 - sync interface.

Index