United Electronic Industries

The High-Performance Alternative

DNx-AI-255

—

User Manual

2-Channel Synchro/Resolver I/O Interface Board
for the PowerDNA Cube and RACK Series Chassis

USPTO Patent: 7,957,942

February 2021

PN Man-DNx-AI-255

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 this Manual 1

1.2 AI-255 Board Overview 3
1.2.1 Synchro / Resolver Inputs & Simulated Outputs 3
1.2.2 Compatibility 3
1.2.3 Environmental Conditions 3
1.2.4 Software Support 3

1.3 Features 4

1.4 Specification 4

1.5 Indicators 7

Chapter 2 AI-255 Synchro / Resolver Functional Description 8

2.1 Synchro & Resolver Overview 8
2.1.1 Synchros 8
2.1.2 Resolvers 12
2.1.3 Example Waveforms 14

2.2 AI-255 Device Architecture 18

2.3 AI-255 Operational Modes 19
2.3.1 Synchro Modes 19
2.3.2 Resolver Modes 20

2.4 Pinout 21

2.5 Connecting Synchro / Resolver Hardware 22
2.5.1 Synchro Wiring. 22
2.5.2 Resolver Wiring 23
2.5.3 Synchro Line-to-Line Voltage. 24
2.5.4 Z-grounded Mode 26
2.5.5 Troubleshooting 27

Chapter 3 Programming with the High-level API 28

3.1 About the High-level Framework. 28

3.2 Creating a Session 28

3.3 Configuring the Resource String. 28

3.4 Configuring for Input 29

3.5 Configuring for Simulated Synchro/Resolver Output 30

3.6 Configuring the Timing 30

3.7 Reading Data 31

3.8 Writing Data 31

3.9 Cleaning-up the Session. 31

Chapter 4 Programming with the Low-level API 32

4.1 About the Low-level API 32

4.2 Low-level Functions 32

4.3 Low-level Programming Techniques. 33
4.3.1 Data Collection Modes 33
4.3.2 Configuring Excitation 34
4.3.3 Configuring the Operational Mode 36
4.3.4 Enabling AI-255 Channels 37
4.3.5 Reading Synchro or Resolver Inputs (Immediate Mode) 37
4.3.6 Writing Synchro or Resolver Simulated Outputs (Immediate Mode) 38

Appendix A 39

A.1 Accessories 39

Appendix B 40

B.1 Synchro Input Mode with Internal Excitation 41

B.2 Synchro Input Mode with External Excitation. 43

B.3 Synchro Simulator Output Mode with Internal Excitation. 45

B.4 Synchro Simulator Output Mode with External Excitation 47

B.5 Synchro Simulator Output Mode with External Excitation & Z-grounding 49

B.6 Resolver Input Mode with Internal Excitation. 50

B.7 Resolver Input Mode with External Excitation 51

B.8 Resolver Simulator Output Mode with Internal Excitation 52

B.9 Resolver Simulator Output Mode with External Excitation. 53

List of Figures

Chapter 1 Introduction 1
1-1 Photo of DNA-AI-255 Synchro/Resolver Board 7

Chapter 2 AI-255 Synchro / Resolver Functional Description 8
2-1 Internal Structure of Star Synchro (left) vs. Delta Synchro (right)..........8
2-2 Synchro Transmitter and Receiver System 10
2-3 Synchro Transmitter and Control Transformer System 11
2-4 Brushless Resolver Control Transformer..........12
2-5 Magnitudes of SIN and COS Output RMS Voltages vs. Rotor Angle θ..........13
2-6 SIN and COS Output Voltages vs. Rotor Angle 14
2-7 AI-255 Resolver Waveforms at 30° Rotor Angle 15
2-8 AI-255 Resolver Waveforms at 45° Rotor Angle 15
2-9 AI-255 Resolver Waveforms at 135° Rotor Angle 16
2-10 AI-255 Synchro Waveforms at -30° Rotor Angle 16
2-11 Block Diagram of DNx-AI-255 I/O Board 18
2-12 Pinout Diagram for DNx-AI-255 21
2-13 Peak-to-peak Voltage Measurement of Synchro 24
2-14 Connecting Star Synchro in Z-grounded Mode 26

Chapter 3 Programming with the High-level API 28

Chapter 4 Programming with the Low-level API 32

Appendix A 39
A-1 Pinout and photo of DNA-STP-62 screw terminal panel..........39

Appendix B 40
B-1 Star Synchro: Input Mode, Internal Excitation 41
B-2 Delta Synchro: Input Mode, Internal Excitation 42
B-3 Star Synchro: Input Mode, External Excitation 43
B-4 Delta Synchro: Input Mode, External Excitation 44
B-5 Star Synchro: Simulator Mode, Internal Excitation 45
B-6 Delta Synchro: Simulator Mode, Internal Excitation 46
B-7 Star Synchro: Simulator Mode, External Excitation..........47
B-8 Delta Synchro: Simulator Mode, External Excitation 48
B-9 Star Synchro: Simulator Mode, External Excitation, Z-grounding..........49
B-10 Resolver: Input Mode, Internal Excitation 50
B-11 Resolver: Input Mode, External Excitation..........51
B-12 Resolver: Simulator Mode, Internal Excitation..........52
B-13 Resolver: Simulator Mode, External Excitation..........53

List of Tables

Chapter 1 Introduction . . . 1
1-1 DNx-AI-255-1 Technical Specifications ... 5
1-2 DNx-AI-255-815 Technical Specifications ... 6
1-3 AI-255 Indicators ... 7

Chapter 2 AI-255 Synchro / Resolver Functional Description . . . 8
2-1 AI-255 Synchro Operational Modes ... 19
2-2 AI-255 Resolver Operational Modes ... 20
2-3 Synchro Connections for Operational Modes ... 22
2-4 Resolver Connections for Operational Modes ... 23
2-5 Z-grounded modes of operation from powerdna.h ... 26

Chapter 3 Programming with the High-level API . . . 28

Chapter 4 Programming with the Low-level API . . . 32
4-1 Summary of Low-level API Functions for DNx-AI-255 ... 32
4-2 AI-255 Modes of Operation ... 36

Appendix A . . . 39

Appendix B . . . 40

Chapter 1 Introduction

This document outlines the feature set and use of the DNx-AI-255, a 2-channel synchro/resolver interface board.

The following product versions are described in this manual:

- DNx-AI-255-1: signal inputs at 2-28 V_{rms}
- DNx-AI-255-815: signal inputs at 5-115 V_{rms}
  (external excitation required if operating at greater than 28 V_{rms})

The following sections are provided in this chapter:

- Organization of this Manual (Section 1.1)
- AI-255 Board Overview (Section 1.2)
- Features (Section 1.3)
- Specification (Section 1.4)
- Indicators (Section 1.5)

1.1 Organization of this Manual

The AI-255 User Manual is organized as follows:

- **Introduction**
  Chapter 1 provides an overview of DNx-AI-255 features, device architecture, connectivity, and logic.
- **AI-255 Synchro / Resolver Functional Description**
  Chapter 2 provides an overview of the how synchros and resolvers work, example waveforms, the AI-255 device architecture, descriptions of the AI-255 modes of operations, and the AI-255 pinout, along with pinouts for connecting to synchros or resolvers.
- **Programming with the High-Level API**
  Chapter 3 provides an overview of the how to create a session, configure the session, and interpret results with the Framework API.
- **Programming with the Low-Level API**
  Chapter 4 is an overview of low-level API commands for configuring and using the AI-255 series board.
- **Appendix A - Accessories**
  This appendix provides a list of accessories available for use with the DNx-AI-255 Synchro/Resolver interface board.
- **Appendix B - Connection Diagrams**
  This appendix contains connection diagrams for various operating and synchro/resolver excitation modes of the DNx-AI-255 interface board.
- **Index**
  This is an alphabetical listing of the topics covered in this manual.

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

Manual Conventions

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

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

NOTE: Notes alert you to important information.

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

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

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

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

Examples of Manual Conventions

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

Usage of Terms

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

1.2 AI-255 Board Overview

The DNx-AI-255 can act as a 2-channel synchro or resolver input or simulator output interface. It is suited for a wide variety of industrial, military, and simulator applications.

The DNA-AI-255, DNR-AI-255, and DNF-AI-255 boards 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 Synchro / Resolver Inputs & Simulated Outputs

The AI-255 board provides two channels that can monitor either 3/4-wire (plus excitation) synchros or 4-wire (plus rotor excitation) resolvers or, as an alternative, provide simulated outputs for test and simulation applications. It is capable of angle measurement accuracies approaching ±2.6 arc-minutes. Each channel may be configured either as an input or an output, in any combination. Output accuracy is ±4 arc-minutes.

The inputs may be sampled at rates up to the excitation frequency of 4 kHz. Each channel provides its own programmable excitation voltage with outputs independently programmable from 2 to 28 V_{rms} at 1.2 VA and at frequencies from 50 to 4000 Hz. When using external excitation, the AI-255 automatically adjusts simulated outputs for variable amplitude and frequency excitations in one reference cycle.

Each DNx-AI-255 channel can be programmed as a synchro or resolver output, ideal for driving devices such as attitude indicators or as test sources for a wide range of synchro or resolver input devices. The two outputs each can accept an independent excitation signal and offer 16-bit output resolution. Each channel can drive up to 28 V_{rms} at 1.2 VA without external buffering.

1.2.2 Compatibility

The AI-255 functionality is similar to the AI-256, and uses the same subset of Synchro/Resolver software functions as the AI-256, but operates at lower frequency, lower currents and higher voltages than the AI-256.

1.2.3 Environmental Conditions

As with all UEI PowerDNA boards, the DNx-AI-255 can be operated in harsh environments and has been tested at 5*g* vibration, 50*g* shock, -40 to +85°C temperature, and altitudes up to 70,000 feet. Each board provides 350 V_{rms} isolation between channels and also between the board and its enclosure or any other installed boards as well as electro-shock-discharge (ESD) isolation.

1.2.4 Software Support

Software included with the DNx-AI-255 provides a comprehensive yet easy to use API that supports all popular operating systems including Windows, Linux, real-time operating systems such as QNX, RTX, VxWorks and more. The UEIDAQ framework comes with bindings for various programming languages such as C, C++, C#, VB.NET and scientific software packages such as LabVIEW and Matlab, as well as supporting OPC servers.

1.3 Features

The features of the DNx-AI-255 include:

- Two Synchro or Resolver input or output channels in any combination
- 16-bit resolution
- 3- / 4-wire (plus excitation) Synchro and 4-wire (plus excitation) Resolver inputs
- Excitation output (28 V_{rms}) for each channel with 1.2 mV_{rms} resolution
- 2-28 V_{rms} inputs on the AI-255-1; 5-115 V_{rms} inputs on the AI-255-815 (external excitation required if operating at greater than 28Vrms)
- User-programmable Excitation frequency range of 50 Hz to 4 kHz (±0.5%) for each channel
- Up to 28 V_{rms} output/excitation at 1.2 VA without external buffering
- Isolation up to 350 V_{rms} between channel and between I/Os and GND
- Weight of 136 g or 4.79 oz for DNA-AI-255; 817 g or 28.8 oz with PPC5.
- Tested to withstand 5g Vibration, 50g Shock, -40 to +85°C Temperature, and Altitude up to 70,000 ft or 21,000 meters.

1.4 Specification

The technical specifications for the DNx-AI-255-1 board are listed in **Table 1-1**.

Technical specifications for the DNx-AI-255-815 board are listed in **Table 1-2**.

Table 1-1. DNx-AI-255-1 Technical Specifications

Inputs	
Number of channels	2
Configuration	Synchro (3-wire) or Resolver (4-wire) may be selected via software
Resolution	16-bit
Accuracy	± 2.6 arc-minute
Frequency	50 Hz to 4.0 kHz
Signal Inputs	2-28 Vrms.
Input Impedance	478 kΩ ±10 kΩ
Acceleration	300 rps/s @ 60 Hz, 450 rps/s @ 400 Hz 1000 rps/s @ 4000 Hz
Step response	800 mS - 179° @ 60 Hz, 150 mS - 179° @ 2500 Hz
Update rate	Maximum update rate is equal to the excitation frequency.
Reference output	
Number of channels	2 (one per input channel)
Output voltage	28 Vrms up to 1.2 VA.
Voltage resolution	1.2 mVrms
Reference Frequency	50 Hz to 4 kHz (±0.5%)
Synchro / Resolver Outputs	
Number of channels	2 (total number of synchro/resolver inputs and simulated outputs is limited to 2.)
Configuration	Synchro (3-wire) or Resolver (4-wire)
Resolution	16-bit
Output Voltage	28 Vrms up to 1.2 VA.
Output Accuracy	±4 arc-minutes
General Specifications	
Operating temperature	Tested -40 °C to +85 °C (for operation above 60 °C in non GigE Cubes the DNA-FAN is required.)
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*	100 g, 3 ms half sine, 18 shocks @ 6 orientations 30 g, 11 ms half sine, 18 shocks @ 6 orientations
Humidity	5 to 95%, non-condensing
Altitude	0 to 70,000 feet
MTBF	275,000 hours
Power consumption	4.5 Watt at idle, up to 10W at full load

Table 1-2. DNx-AI-255-815 Technical Specifications

Inputs	
Number of channels	2
Configuration	Synchro (3-wire) or Resolver (4-wire) may be selected via software
Resolution	16-bit
Accuracy	± 2.6 arc-minute
Frequency	50 Hz to 4.0 kHz
Signal Inputs	5-115 Vrms. (external reference required if operating at greater than 28 Vrms)
Acceleration	300 rps/s @ 60 Hz 450 rps/s @ 400 Hz 1000 rps/s @ 4000 Hz
Step response	800 mS - 179° @ 60 Hz 150 mS - 179° @ 2500 Hz
Update rate	Maximum update rate is equal to the excitation frequency.
Reference output	
Number of channels	2 (one per input channel)
Output voltage	28 Vrms up to 1.2 VA.
Voltage resolution	1.2 mVrms
Reference Frequency	50 Hz to 4 kHz (±0.5%)
Synchro / Resolver Outputs	
Number of channels	2 (total number of synchro/resolver inputs and simulated outputs is limited to 2.)
Configuration	Synchro (3-wire) or Resolver (4-wire)
Resolution	16-bit
Output Voltage	28 Vrms up to 1.2 VA.
Output Accuracy	±4 arc-minutes
General Specifications	
Operating temperature	Tested -40 °C to +85 °C (for operation above 60 °C in non GigE Cubes the DNA-FAN is required.)
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*	100 g, 3 ms half sine, 18 shocks @ 6 orientations 30 g, 11 ms half sine, 18 shocks @ 6 orientations
Humidity	5 to 95%, non-condensing
Altitude	0 to 70,000 feet
MTBF	275,000 hours
Power consumption	4.5 Watt at idle, up to 10W at full load

1.5 Indicators

The DNx-AI-255 indicators are described in **Table 1-3** and illustrated in **Figure 1-1**.

Table 1-3. AI-255 Indicators

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

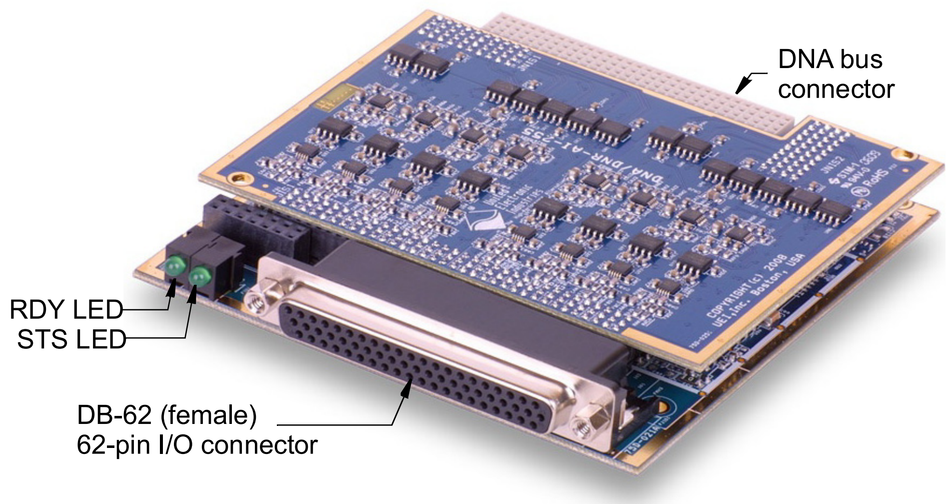


Figure 1-1 Photo of DNA-AI-255 Synchro/Resolver Board

Chapter 2 AI-255 Synchro / Resolver Functional Description

The DNx-AI-255 can act as a 2-channel Synchro or Resolver input interface or simulated output interface for UEI data acquisition systems.

This chapter provides the following information:

- Synchro & Resolver Overview (Section 2.1)
- AI-255 Device Architecture (Section 2.2)
- AI-255 Operational Modes (Section 2.3)
- Pinout (Section 2.4)
- Connecting Synchro / Resolver Hardware (Section 2.5)

2.1 Synchro & Resolver Overview

Synchros and resolvers are electromechanical transducers that are used either to detect and measure a rotary shaft position or to position a shaft at a desired angle. The devices can be further classified as transmitters, receivers, differentials, or control transformers.

2.1.1 Synchros

A synchro consists of a single-phase rotor surrounded by a 3-phase stator. The three stator coils are spatially arranged 120° apart but may be wired in either a star or delta configuration, as illustrated in **Figure 2-1**.

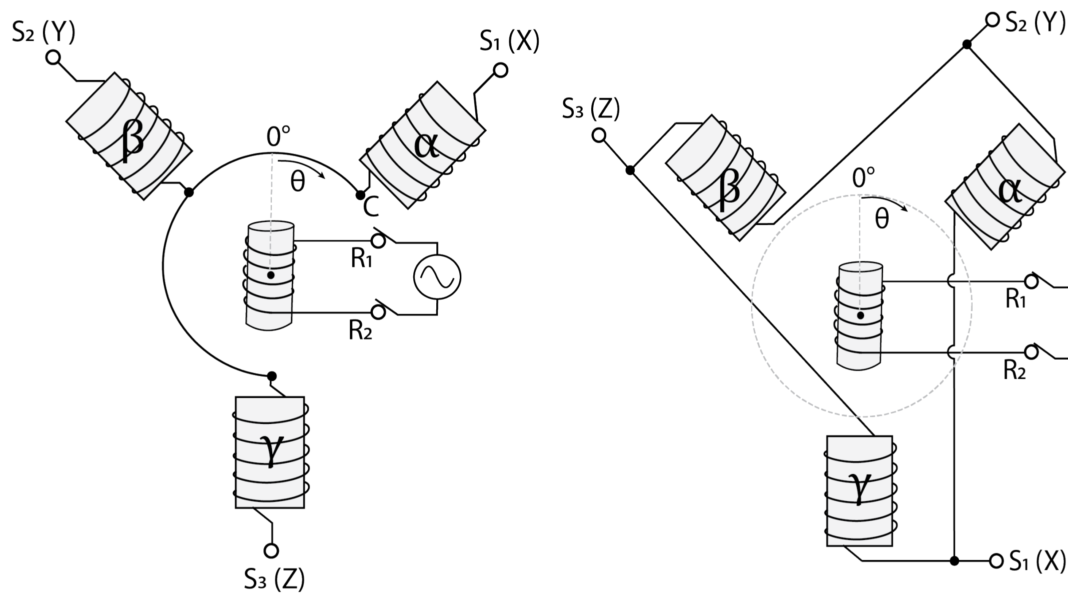


Figure 2-1 Internal Structure of Star Synchro (left) vs. Delta Synchro (right)

NOTE: Synchro stator terminals are commonly labeled X, Y, Z in practice, but this manual will use the standard S_1, S_2, S_3 notation moving forward.

Although synchros may appear similar in construction to synchronous motors/generators, the main difference between them is that the rotor of a synchro is excited with an AC voltage rather than a DC voltage. Since the magnetic field created by the synchro's rotor changes with time, voltage is induced in the stator even when the rotor is stationary. This allows a synchro transmitter to monitor the rotor's angular position, unlike a generator where rotor motion is required to produce an output.

Transmitter

A synchro transmitter applies AC voltage to the rotor winding and induces AC voltages in the three stator windings. The induced voltage is maximized when the rotor and stator coil share an axis and minimized when the axes are perpendicular. In general, the amplitudes are given by:

$$V_\alpha = kV_R \sin(\theta + 30°)$$
$$V_\beta = kV_R \sin(\theta + 150°)$$
$$V_\gamma = kV_R \sin(\theta + 270°)$$

where

- θ is the rotor angle measured clockwise from the reference position shown in **Figure 2-1**.
- V_α is the (time-varying) voltage across stator coil α.
  - Star Synchro: $V_\alpha = V_{S1} - V_C$
  - Delta Synchro: $V_\alpha = V_{S2} - V_{S1}$
- V_R is the (time-varying) excitation voltage across the rotor.
  $V_R = V_{R1} - V_{R2}$
- k is a constant which represents the maximum coupling transformation ratio V_{out}/V_{in}.

Thus, the transmitter outputs a unique set of three voltages for each position of the rotor throughout a 360° rotation. If the sine evaluates to a positive number, the induced voltage is in time-phase with the excitation voltage. If the sine is negative, the induced voltage is inverted (180° out of time-phase). Example waveforms are shown in **Figure 2-10**.

NOTE: If the device uses a different reference shaft position from what's shown in **Figure 2-1**, θ will be offset by a constant.

Receiver

A synchro receiver has the same physical construction as a synchro transmitter. However, instead of functioning as an output device, the receiver moves its rotor in response to input signals present on its stator windings.

In a typical application, a synchro transmitter and receiver are wired together so that the angular position of the transmitter's rotor is automatically reproduced in the receiver's rotor. As shown in **Figure 2-2**, the receiver's rotor is excited in parallel with the transmitter. When the transmitter and receiver rotors are in alignment, stator voltages are equal and no current flows. If the transmitter rotor is turned (relative to the receiver rotor), a force appears in the receiver, causing the rotor to track the transmitter rotor. The torque produced is proportional to the angle difference between the two rotors. Typical accuracy of such a system is 30 arc-minutes.

A single transmitter may be parallel-connected to multiple receivers, at the cost of reducing accuracy and increasing power drain from the source.

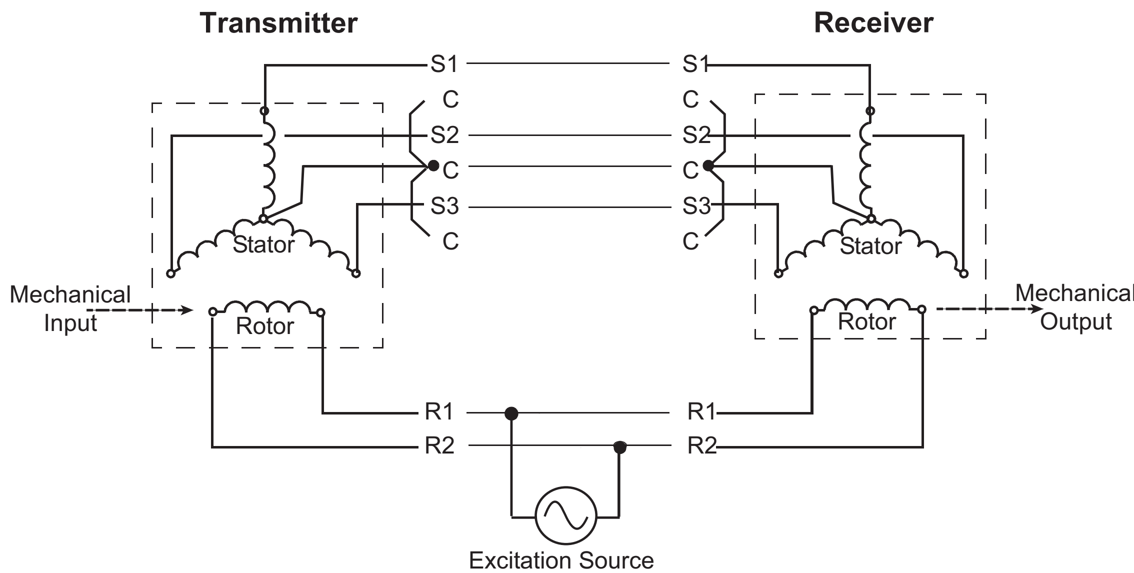


Figure 2-2 Synchro Transmitter and Receiver System

Control Transformer

A control transformer is another type of synchro that can be wired to a transmitter. Unlike a receiver, the control transformer is not designed to directly position a load. Instead, its rotor terminals are normally wired to a servo control system that can drive higher torque loads.

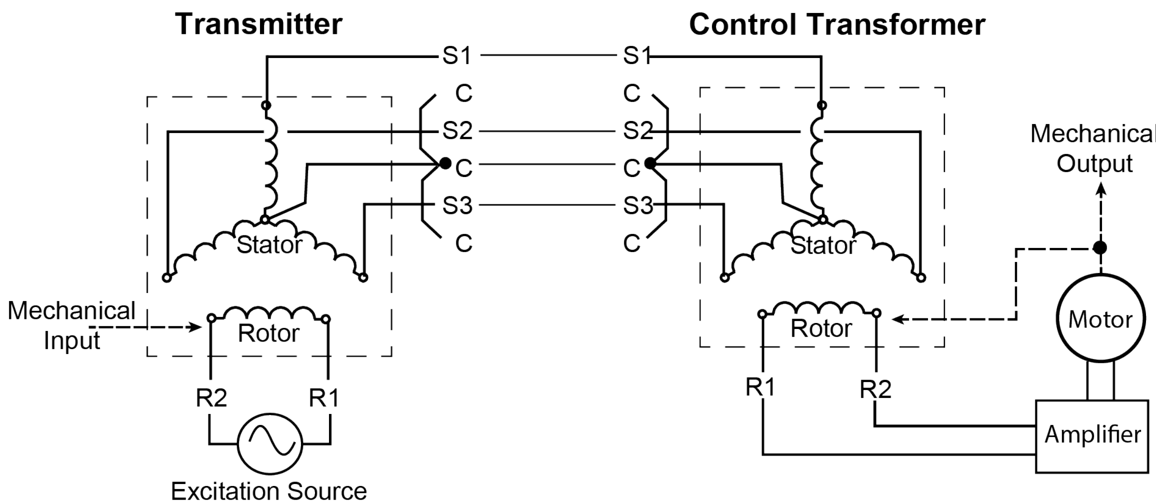


Figure 2-3 Synchro Transmitter and Control Transformer System

When the transmitter rotor is turned, the transmitter outputs a new set of stator voltages to the control transformer. This changes the magnetic field seen by the control transformer rotor and induces a voltage if the two rotors are not aligned. The voltage induced in the control transformer rotor is proportional to the sine of the angle difference between the two rotors. (The control transformer rotor is cylindrically shaped to keep the magnitude of its stator field constant, allowing the induced voltage to depend only on the field direction.) Therefore, the control transformer provides information about the transmitter rotor's angular position.

In a typical application, the small but sensitive output signal from the control transformer is amplified and sent to a motor. As the motor positions the load, it mechanically turns the control transformer shaft in a negative feedback loop; when the control transformer reaches the same angle as the transmitter, the input to the motor decreases back to 0.

2.1.2 Resolvers

A resolver is a rotary transformer in which the magnitude of the energy through the resolver varies sinusoidally with rotation of the shaft. A resolver control transmitter has one primary winding (Excitation Winding) and two secondary windings (the SIN and COS windings). The excitation winding is located on the rotor and the SIN and COS windings are on the stator, displaced spatially by 90°. If the resolver is a brushless type, current is applied through a rotary transformer, which eliminates the problems of slip rings and brushes.

The connection arrangement of a brushless resolver control transformer is illustrated below in **Figure 2-4**.

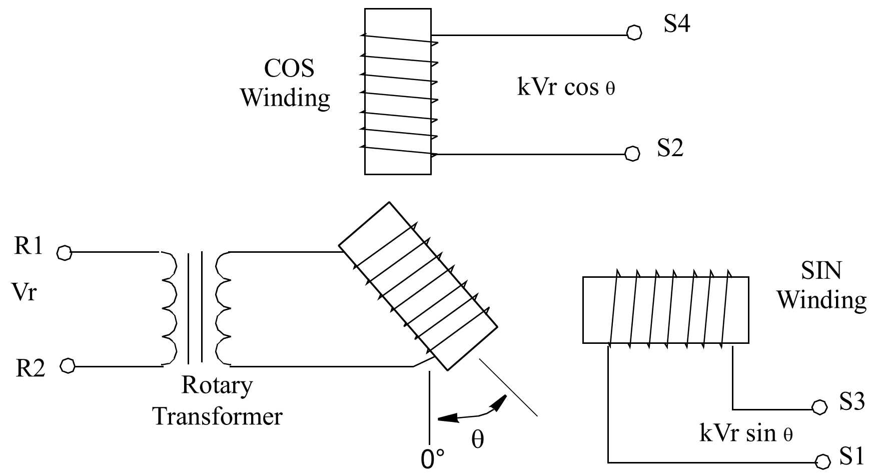


Figure 2-4 Brushless Resolver Control Transformer

The excitation winding on the rotor is typically excited by an AC voltage. The induced voltages in the SIN and COS windings obey the following equations:

$$V_{SIN} = kV_R \sin(\theta)$$
$$V_{COS} = kV_R \cos(\theta)$$

where

- θ is the rotor angle relative to the reference position shown in **Figure 2-4**.
- V_{SIN} is the (time-varying) voltage across the SIN winding.

  $V_{SIN} = V_{S1} - V_{S3}$
- V_{COS} is the (time-varying) voltage across the COS winding.

  $V_{COS} = V_{S2} - V_{S4}$
- V_R is the (time-varying) excitation voltage applied to the rotor.

  $V_R = V_{R1} - V_{R2}$
- k is a constant which represents the maximum coupling transformation ratio V_{out}/V_{in}.

Figure 2-5 illustrates how the magnitudes of the output voltages vary with rotor angle.

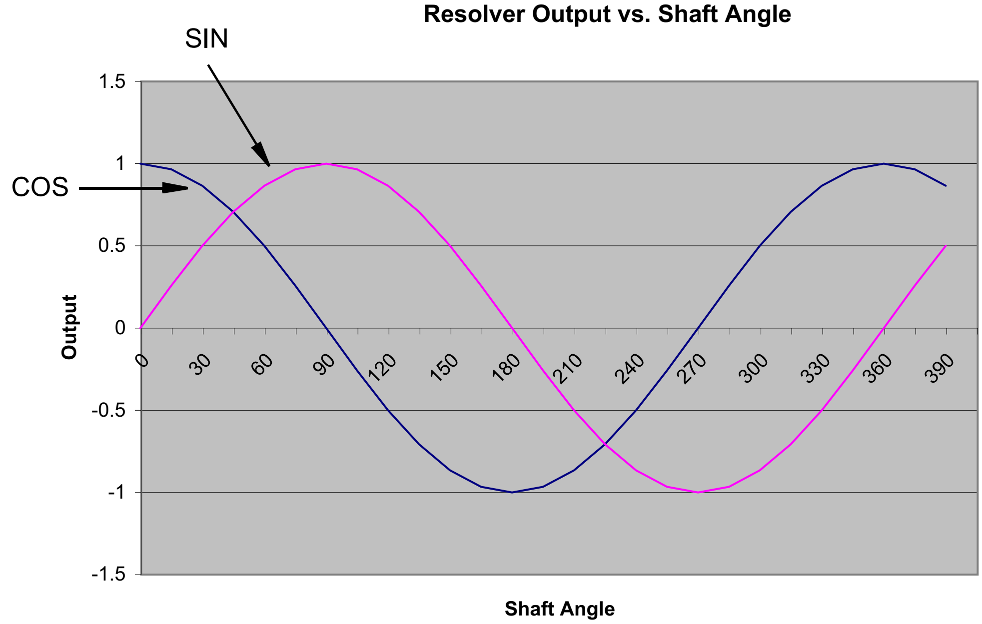


Figure 2-5 Magnitudes of SIN and COS Output RMS Voltages vs. Rotor Angle θ

The induced voltage is maximized when the primary and secondary windings share an axis and minimized when the axes are perpendicular. For example, at θ=0° (first point on the graph), the rotor is in line with the COS winding. Since cos(0°)=1, V_{COS} will have the maximum possible magnitude. Since sin(0°)=0, there will be no output on V_{SIN}. If the rotor is positioned at θ=45°, the SIN and COS output waveforms will have equal magnitudes.

Note that the above graph in **Figure 2-5** does not represent an oscilloscope display. Refer to **Figure 2-7**, **Figure 2-8** and **Figure 2-9** for oscilloscope displays of the SIN (S1), COS (S2), and excitation waveforms at rotor angles 30°, 45° and 135°.

2.1.3 Example Waveforms

The following subsections show example waveforms for amplitudes, polarity, and phase vs rotor angles for synchro and resolver circuits.

2.1.3.1 Resolver Waveforms

A sine wave AC Excitation voltage V_{exc} is applied between R1 and R2. The voltage observed between S1 and S3 is $V_{sin} = V_{exc}$ sin A, where A is the rotor angle in radians. Similarly, the voltage observed between S2 and S4 is $V_{cos} = V_{exc}$ cos A, where A is the rotor angle in radians. The two output voltages remain in phase with each other relative to the excitation voltage, but differ in magnitude and/or polarity (relative to excitation) as the rotor angle changes, as shown in **Figure 2-6**.

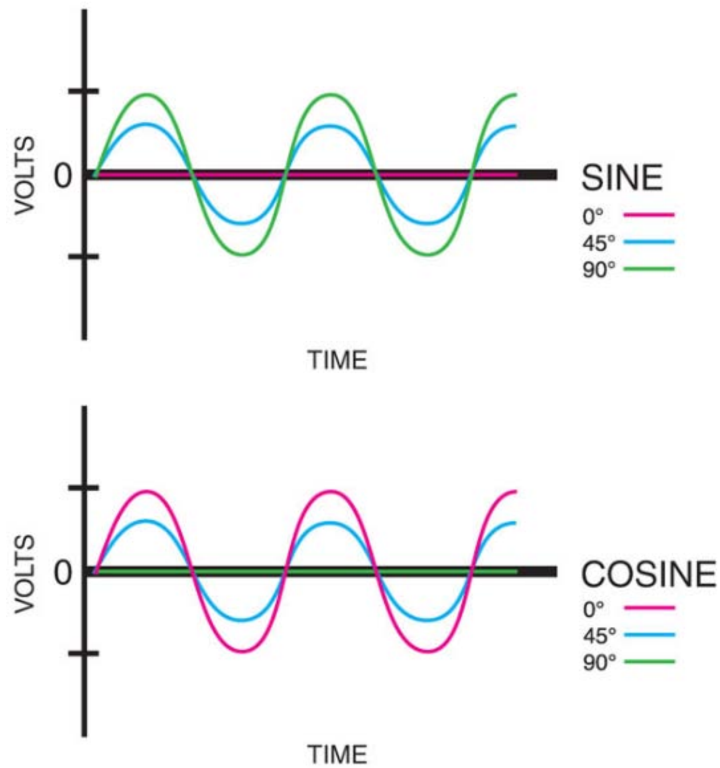


Figure 2-6 SIN and COS Output Voltages vs. Rotor Angle

The output waveforms in resolver simulation mode for the AI-255 are shown at 30°, 45°, and 135° in **Figures 2-7**, **2-9**, and **2-8** respectively.

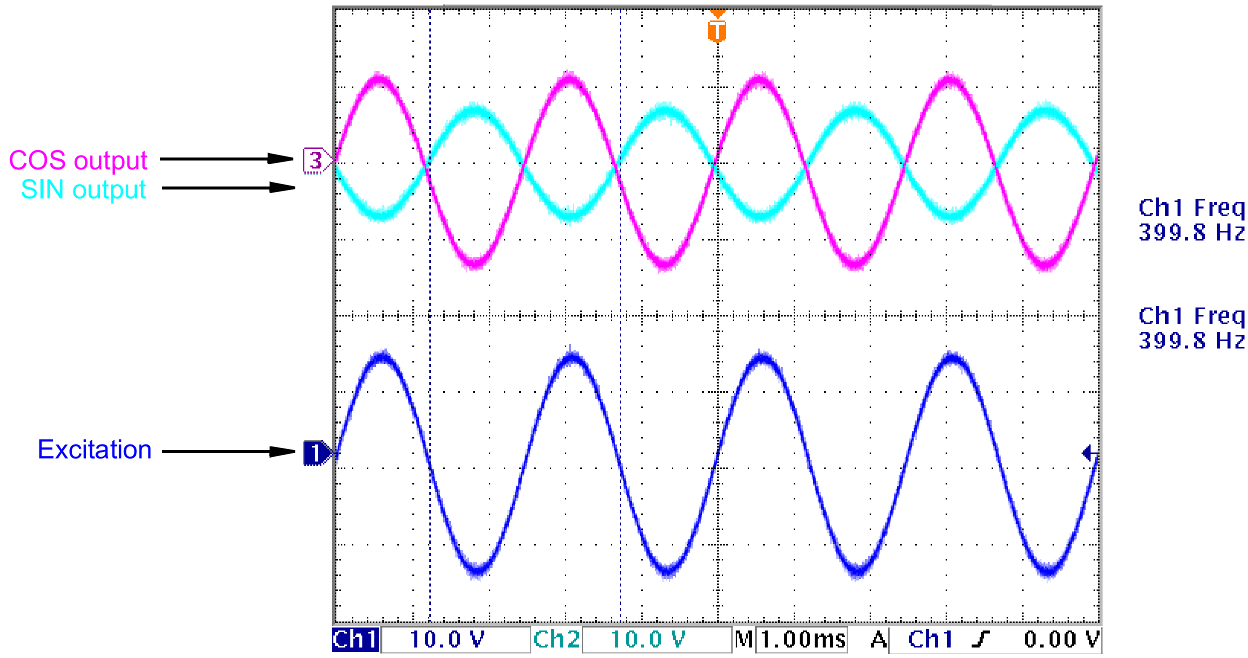


Figure 2-7 AI-255 Resolver Waveforms at 30° Rotor Angle

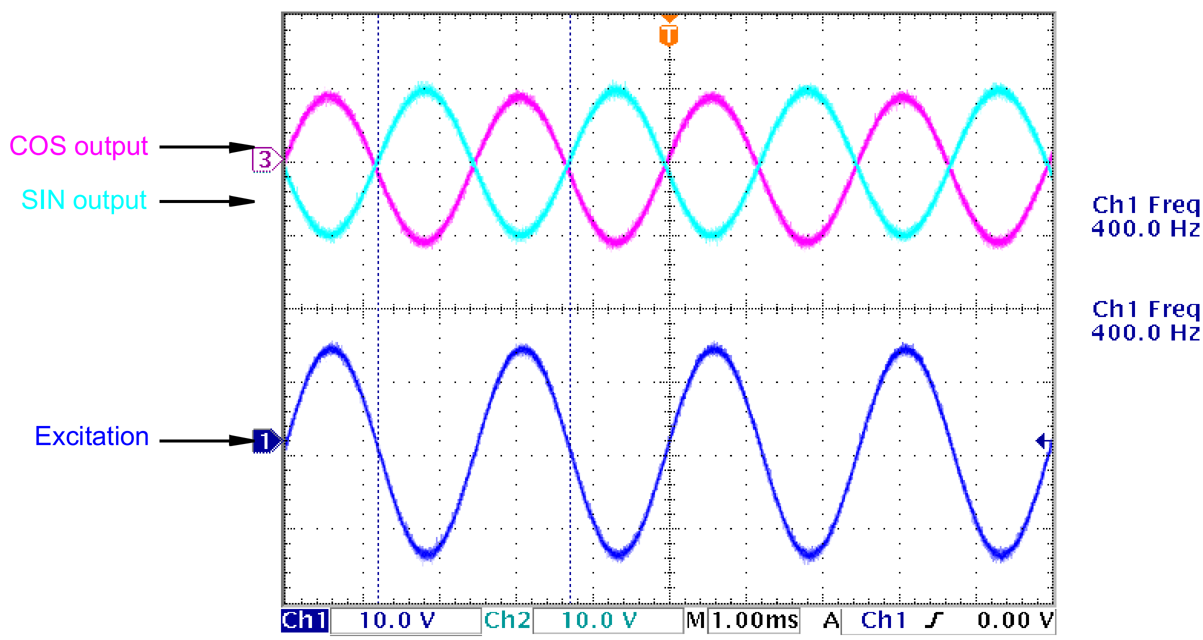


Figure 2-8 AI-255 Resolver Waveforms at 45° Rotor Angle

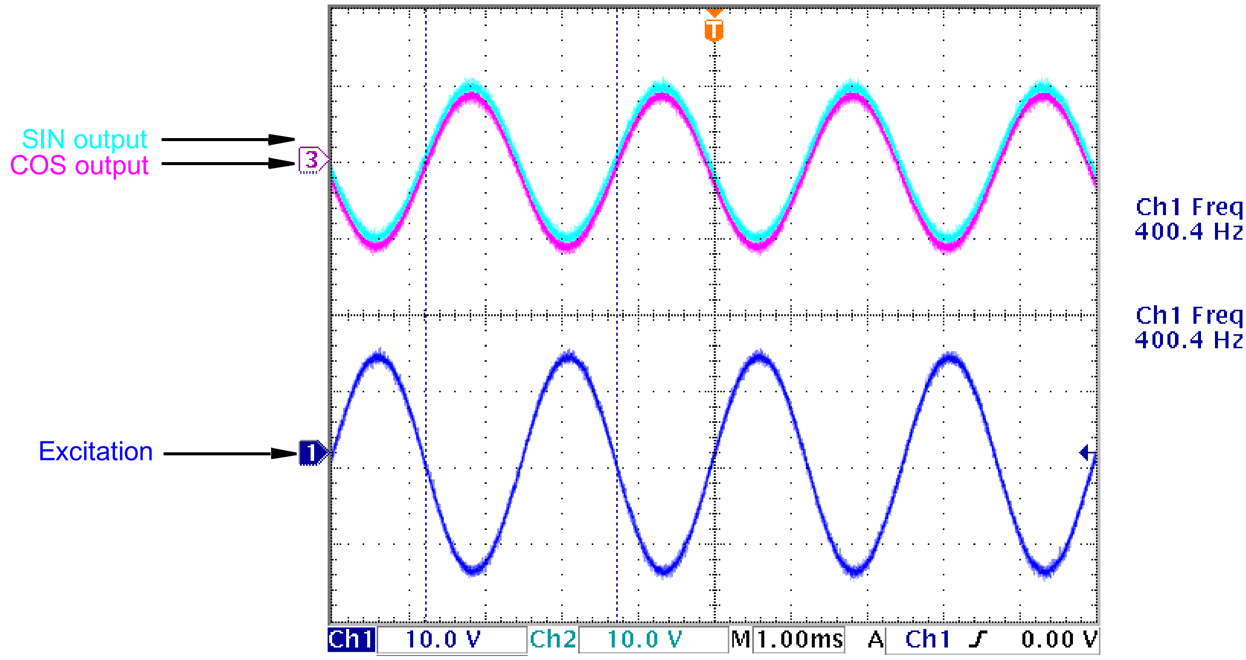


Figure 2-9 AI-255 Resolver Waveforms at 135° Rotor Angle

2.1.3.2 Synchro Waveforms

When a synchro is used, the excitation and output voltages appear as shown in **Figure 2-10**. Note that a synchro has three windings with angles between coils of 120°.

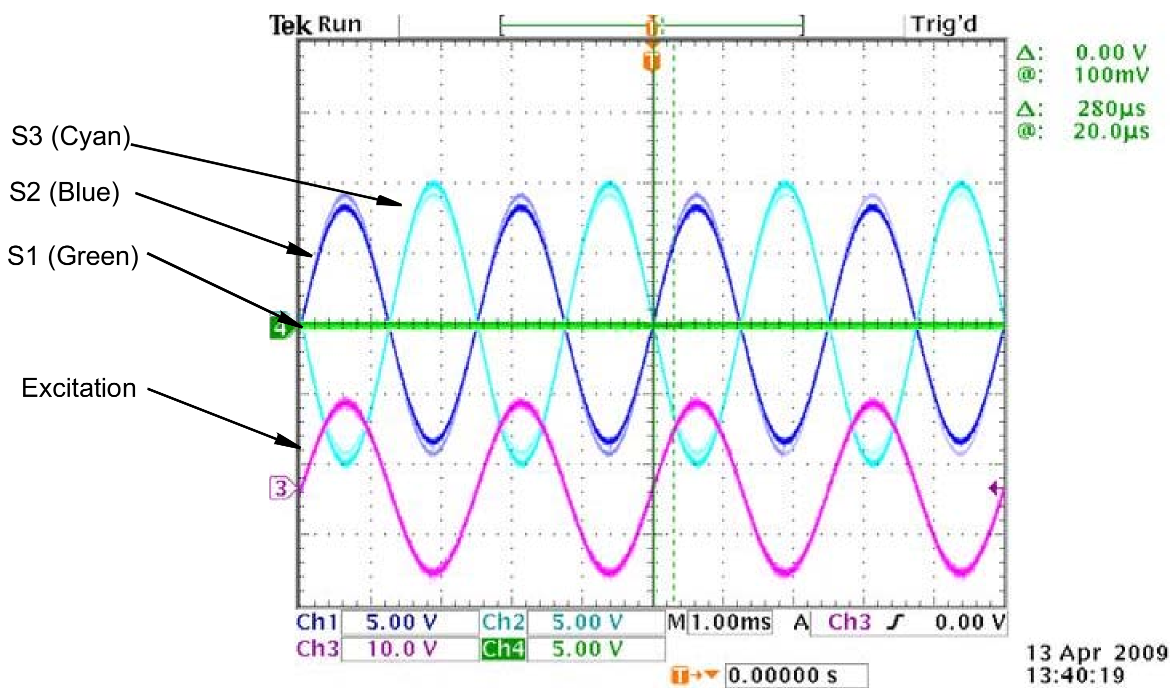


Figure 2-10 AI-255 Synchro Waveforms at -30° Rotor Angle

The S1 coil output voltage (green line) is zero because the rotor is positioned plus or minus 90° relative to the S1 stator winding and therefore produces nothing. The S2 coil (blue line) shows a voltage in phase with excitation and with the same polarity as the excitation voltage. The S3 coil (cyan line) shows a voltage of polarity opposite to that of the excitation (or 180° out of phase).

NOTE: Coils on a synchro can be labeled in two different ways. Looking at the synchro from the shaft side with S1 at the top, S2 and S3 may follow in either a counterclockwise or clockwise direction depending on the manufacturer.

NOTE: When using the AI-255 as a synchro output for simulation, you can attach a scope to the simulation outputs, ground the scope probes to AGND and read voltages between S1 and AGND, S2 and AGND, and S3 and AGND. Some synchros have the coil mid points between coils brought out, but most do not.

Additional Information

Additional information and formulas for calculating the simulated positions are provided in the PowerDNA API Reference Manual. Refer to the DqAdv255Write and DqAdv255ConvertSim API.

2.2 AI-255 Device Architecture

A block diagram of a DNx-AI-255 board is shown in **Figure 2-11**.

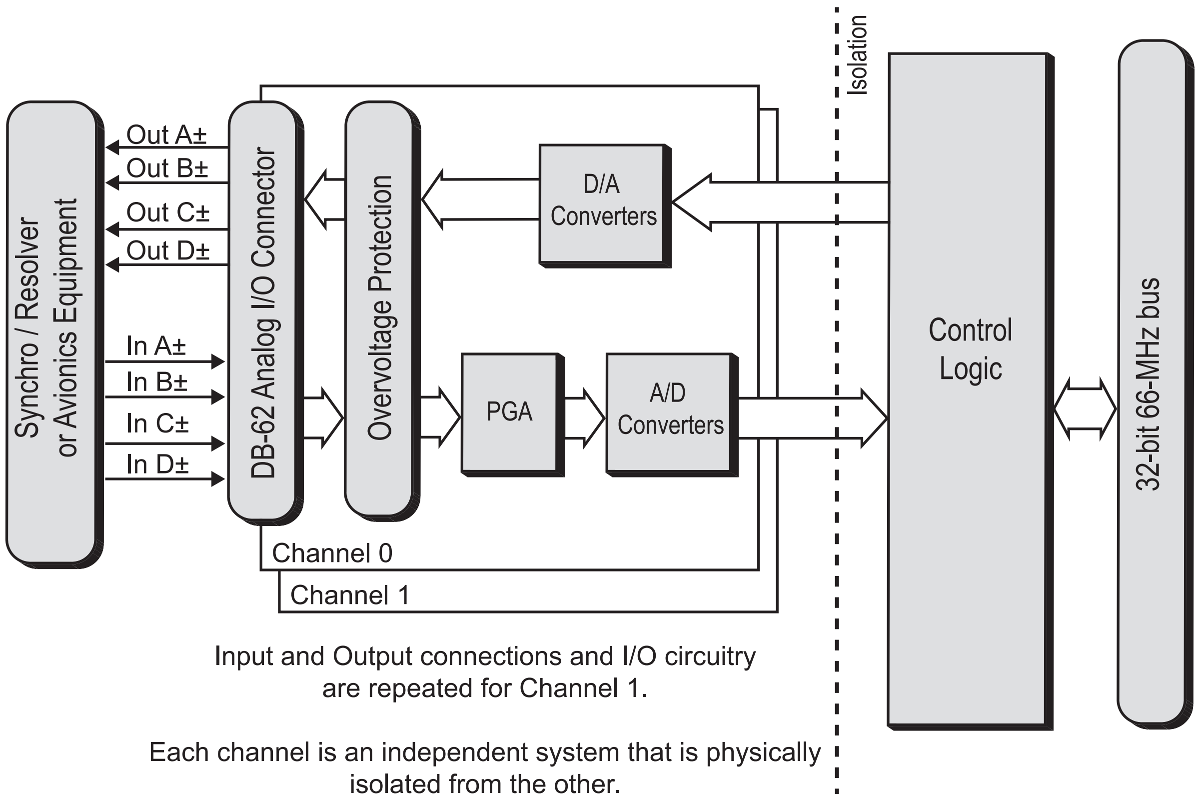


Figure 2-11 Block Diagram of DNx-AI-255 I/O Board

Board logic is divided into isolated and non-isolated sections. The isolated side handles all functions associated with the sensor input and output circuits, and the non-isolated side handles all Cube or chassis-related operations.

As shown in **Figure 2-11**, the AI-255 provides two Synchro/Resolver interfaces (channels). The analog inputs from a synchro or resolver sensor are routed as differential inputs through the DB connector, through over protection circuitry to a programmable gain amplifier (PGA). A user-programmed gain and offset are applied. A 16-bit SAR analog to digital converter digitizes the PGA output for logic analysis and calculations.

2.3 AI-255 Operational Modes

This section describes the various operational modes available on the AI-255. For information about programming the board configuration and setting operating modes and parameters, refer to **Chapters 3** and **4**.

2.3.1 Synchro Modes

Table 2-1 lists the four operational modes for star and delta synchros. Each synchro mode also supports a z-grounded configuration (see Section 2.5.4 for more information about z-grounding).

Table 2-1. AI-255 Synchro Operational Modes

Input/Output	Excitation	Description
Synchro Input	Internal excitation	AI-255 reads the voltages on the stator coils as analog inputs and also supplies the excitation voltage to the rotor coil. See Section B-1 on page 41 for hardware connections
Synchro Input	External excitation	AI-255 reads the voltages on the stator coils as analog inputs. An external source supplies the excitation voltage to the rotor coil, which is readback by the AI-255 as an analog input. See Section B-3 on page 43 for hardware connections
Synchro Simulation	Internal excitation	The AI-255 outputs voltages that simulate the analog signals from stator coils of a synchro. It also outputs an analog excitation voltage generated in the AI-255. See Section B-5 on page 45 for hardware connections
Synchro Simulation	External excitation	The AI-255 outputs voltages that simulate the analog signals from stator coils of a synchro. Excitation voltage is supplied by an external source and read back by the AI-255 as an analog input. See Section B-7 on page 47 for hardware connections

2.3.2 Resolver Modes

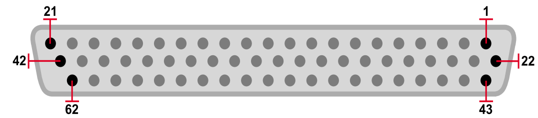
Table 2-2 lists the four operational modes for resolvers.

Table 2-2 AI-255 Resolver Operational Modes

Input/Output	Excitation	Description
Resolver Input	Internal excitation	AI-255 reads the voltages on the stator coils as analog inputs and also supplies the excitation voltage to the rotor coil(s). See Section B-10 on page 50 for hardware connections
Resolver Input	External excitation	AI-255 reads the voltages on the stator coils as analog inputs. An external source supplies the excitation voltage to the rotor coil(s), which is readback by the AI-255 as an analog input. See Section B-11 on page 51 for hardware connections
Resolver Simulation	Internal excitation	The AI-255 outputs voltages that simulate the analog signals from stator coils of a resolver. It also outputs an analog excitation voltage generated in the AI-255. See Section B-12 on page 52 for hardware connections
Resolver Simulation	External excitation	The AI-255 outputs voltages that simulate the analog signals from stator coils of a resolver. Excitation voltage is supplied by an external source and read back by the AI-255 as an analog input. See Section B-13 on page 53 for hardware connections

2.4 Pinout

The pinout of the DNx-AI-255 62-pin DB connector is shown in **Figure 2-12**.



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

Dashed Line represents the isolation barrier between channels

Figure 2-12 Pinout Diagram for DNx-AI-255

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

2.5 Connecting Synchro / Resolver Hardware

This section summarizes the hardware connections between the AI-255 and a synchro or resolver device. Refer to Appendix B for connection diagrams.

2.5.1 Synchro Wiring

Synchro terminals are labeled S1, S2, S3 for the three stator windings, R1/R2 for the rotor, and C for the star synchro center point. Exc+/Exc– refer to the excitation source terminals.

Table 2-3. Synchro Connections for Operational Modes

Signal Name	Pin No.		Input Mode, Internal Excitation		Input Mode, External Excitation		Simulator Mode, Internal Excitation		Simulator Mode, External Excitation	
	Ch 0	**Ch 1**	**Star**	**Delta**	**Star**	**Delta**	**Star**	**Delta**	**Star**	**Delta**
In A+	4	14	S1	S1	S1	S1	NC	NC	NC	NC
In A-	25	35	C	S3	C	S3	NC	NC	NC	NC
In B+	5	15	S3	S2	S3	S2	NC	NC	NC	NC
In B-	26	36	C	S1	C	S1	NC	NC	NC	NC
In C+	9	62	S2	S3	S2	S3	NC	NC	NC	NC
In C-	30	61	C	S2	C	S2	NC	NC	NC	NC
In D+	10	20	NC	NC	Exc+	Exc+	NC	NC	Exc+	Exc+
In D-	31	41	NC	NC	Exc-	Exc-	NC	NC	Exc-	Exc-
OutA+	44	54	NC	NC	NC	NC	S1	S1	S1	S1
OutA-	43	53	NC	NC	NC	NC	C	S3	C	S3
OutB+	2	12	NC	NC	NC	NC	S3	S2	S3	S2
OutB-	23	33	NC	NC	NC	NC	C	S1	C	S1
OutC+	7	17	NC	NC	NC	NC	S2	S3	S2	S3
OutC-	28	38	NC	NC	NC	NC	C	S2	C	S2
OutD+	8	18	R1	R1	NC	NC	R1	R1	NC	NC
OutD-	29	39	R2	R2	NC	NC	R2	R2	NC	NC
NC	24,47	32, 42, 57, 59	--	--	--	--	--	--	--	--
GND	6, 22, 45, 51	16, 34, 40, 55	--	--	--	--	--	--	--	--
Rsvd	1, 3, 27, 46, 48, 49, 50, 52	11, 13, 19, 21, 37, 56, 58, 60	--	--	--	--	--	--	--	--

NOTE: Coils on a synchro can be labeled in two different ways. Looking at the synchro from the shaft side with S1 at the top, S2 and S3 may follow in either a counterclockwise or clockwise direction depending on the manufacturer. Refer to “Troubleshooting” on page 27 if you have questions when connecting synchro stator lines.

2.5.2 Resolver Wiring

Resolver terminals are S1/S3 and S2/S4 for stator windings and R1/R2 for the rotor. Exc+ and Exc– refer to the excitation source terminals.

Table 2-4. Resolver Connections for Operational Modes

Signal Name	Pin No.		Input Mode, Internal Excitation	Input Mode, External Excitation	Simulator Mode, Internal Excitation	Simulator Mode, External Excitation
	Ch 0	Ch 1	Resolver	Resolver	Resolver	Resolver
In A+	4	14	S1	S1	NC	NC
In A-	25	35	S3	S3	NC	NC
In B+	5	15	S2	S2	NC	NC
In B-	26	36	S4	S4	NC	NC
In C+	9	62	NC	NC	NC	NC
In C-	30	61	NC	NC	NC	NC
In D+	10	20	NC	Exc+(R1)	NC	Exc+
In D-	31	41	NC	Exc-(R2)	NC	Exc-
OutA+	44	54	NC	NC	S1	S1
OutA-	43	53	NC	NC	S3	S3
OutB+	2	12	NC	NC	S2	S2
OutB-	23	33	NC	NC	S4	S4
OutC+	7	17	NC(R2)	NC	Opt+(R2)	NC
OutC-	28	38	NC(R4)	NC	Opt-(R4)	NC
OutD+	8	18	R1	NC	Exc+(R1)	NC
OutD-	29	39	R3	NC	Exc-(R3)	NC
NC	24,47	32, 42, 57, 59	--	--	--	--
GND	6, 22, 45, 51	16, 34, 40, 55	--	--	--	--
Rsvd	1, 3, 27, 46, 48, 49, 50, 52	11, 13, 19, 21, 37, 56, 58, 60	--	--	--	--

2.5.3 Synchro Line-to-Line Voltage

The AI-255 measures and simulates the peak-to-peak voltages (V_{pp}) across stator and rotor coils. RACK/Cube logic and firmware computes the corresponding angle from V_{pp} using the equations on page 9, where V_{pp}=2*max(V_{α}).

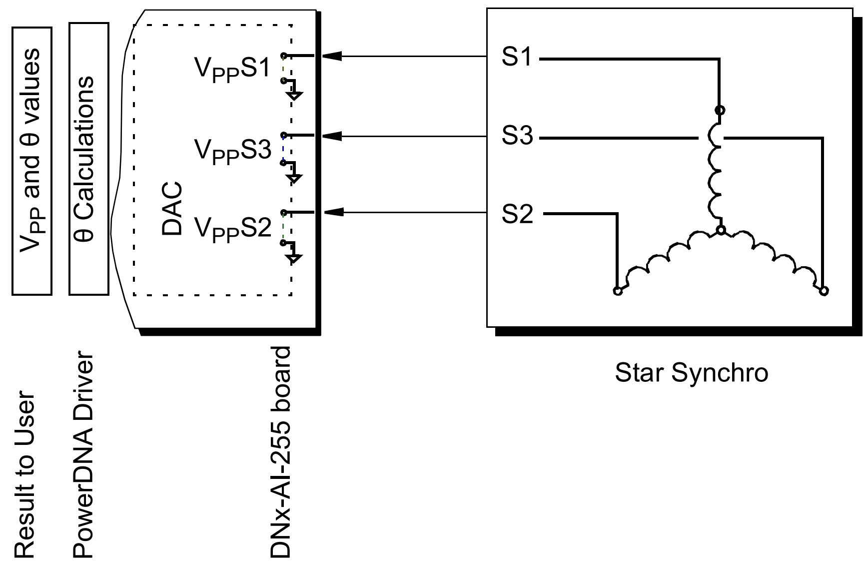


Figure 2-13 Peak-to-peak Voltage Measurement of Synchro

However, synchro documentation will often refer to line-to-line voltages instead of peak-to-peak voltages. A line-to-line voltage is measured across two of the three S_1, S_2, S_3 wires or across R_1, R_2. For delta synchros, the coil voltages are measured differentially and therefore equal the line-to-line voltages. For star synchros, the line-to-line voltages can be derived from the page 9 equations by substituting in the ground-referenced V_{S1}, V_{S2}, and V_{S3} for V_{α}, V_{β}, and V_{γ} respectively.

The resulting line-to-line amplitudes are:

Star Synchro	Delta Synchro
$V_{S1} - V_{S2} = \sqrt{3}kV_R\sin\theta$	$V_{S2} - V_{S1} = kV_R\sin(\theta + 30°)$
$V_{S2} - V_{S3} = \sqrt{3}kV_R\sin(\theta + 120°)$	$V_{S3} - V_{S2} = kV_R\sin(\theta + 150°)$
$V_{S3} - V_{S1} = \sqrt{3}kV_R(t)\sin(\theta + 240°)$	$V_{S1} - V_{S3} = kV_R\sin(\theta + 270°)$

NOTE: If the device uses a different reference shaft position from what's shown in **Figure 2-1**, θ will be offset by a constant.

RMS Voltage

Synchros are most commonly rated using a root mean squared (rms) voltage. When supplying a voltage value to UEI API or interpreting debug waveforms on an oscilloscope, you may need to convert between the max line-to-line RMS voltage amplitude (V_{LL}) from the synchro specification and the max peak-to-peak voltage amplitude (V_{PP}) used by the AI-255 driver.

To convert between V_{LL} and V_{PP}, you can use the following formulas:

Coil	Star Synchro	Delta Synchro
Stator	$V_{PP} = 2\sqrt{2} \times \frac{V_{LL}}{\sqrt{3}} \cong 1.63299V_{LL}$	$V_{PP} = 2\sqrt{2} \times V_{LL} \cong 2.82843V_{LL}$
Rotor	$V_{PP} = 2\sqrt{2} \times V_{LL} \cong 2.82843V_{LL}$	$V_{PP} = 2\sqrt{2} \times V_{LL} \cong 2.82843V_{LL}$

Example:

- A star synchro with the rms excitation voltage of 23.5V will need a ground-referenced peak-to-peak voltage of 66.4 V set for the AI-255.
  66.4V ≈ 2.82843 * 23.5V
- A star synchro's rms stator voltage of 11.8V (between any two stator terminals) will have a maximum peak-to-peak voltage span of 19.26 V.
  19.26V ≈ 1.63299 * 11.8V

Exercise caution when wiring and double-check that correct voltage is set on the AI-255 to avoid overloading and permanently damaging the synchro or resolver.

Low-level Macros

Low-level API includes the following software macros for converting star synchro stator voltages:

- `#define DQ_AI255_RMS_LN_LN_TO_PP(V) ((V)*1.633)`

where input parameter `V` is V_{LL}, and the resulting output is V_{PP}.

- `#define DQ_AI255_RMS_LN_LN_TO_RMS(V) ((V)*0.5774)`

where input parameter `V` is the line-to-line voltage V_{LL}, and the result is the ground-referenced stator voltage in volts RMS (V_{RMS}). The constant 0.5774 is $1/(\sqrt{3})$.

2.5.4 Z-grounded Mode

It is possible to ground the Z (S3) lines of some synchros to the vehicle's common ground to save on wiring - this is called a synchro in *z-grounded mode*.

In hardware, the S3 input/output on the AI-255 is left unconnected for z-grounded synchros. See **Figure 2-14** and Section B-9 on page 49 for connection diagrams.

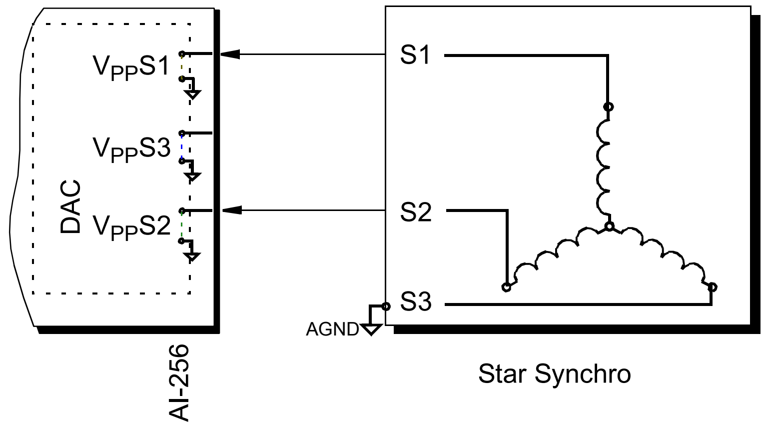


Figure 2-14 Connecting Star Synchro in Z-grounded Mode

Software Modes

Z-grounded delta synchros do not require any software changes since their coil voltages are already measured differentially.

Z-grounded star synchros DO require users to select a special software mode in their AI-255 program:

- When using the Framework API, the enum `UeiSynchroZGroundMode` is used to select z-grounded mode. Refer to **Chapter 3** for more information about programming using the UEIDAQ Framework API.
- When using low-level API, defines listed in the table below are used to set the different z-grounded modes; refer to **Chapter 4** for information on low-level software programming.

Mode of Operation	Description
`DQ_AI255_MODE_SI_INTZ`	Synchro input, internal excitation, Z grounded
`DQ_AI255_MODE_SI_EXTZ`	Synchro input, external excitation, Z grounded
`DQ_AI255_MODE_SS_INTZ`	Synchro output, internal excitation, Z grounded
`DQ_AI255_MODE_SS_EXTZ`	Synchro output, external excitation, Z grounded

Table 2-5. Z-grounded modes of operation from powerdna.h

2.5.5 Trouble-shooting

This section describes some of the conditions observed when the synchro is not wired correctly to the board. Incorrect wiring can be as mild as inaccurate rotor position or as severe as permanently damaging the synchro.

In the mildest cases, the synchro or rotor lines may be in incorrect positions in the terminal panel. Reversing the rotor (V_{ext+}, V_{ext-}) or stator (S1, S2, S3) wires can cause the position of the rotor to be at a wrong angle, or rotate clockwise.

In the more severe cases, the rotor may move in a jerky or erratic manner, the synchro may hum and may be warm/hot to the touch, indicating a possible open connection. Warm or hot synchros may also indicate a short circuit.

The AI-255 board has overvoltage protection up to $350V_{rms}$ and thermal protection; however, the synchro may be permanently damaged by a bad voltage setting. It is recommended to check the configured voltage with an oscilloscope (best set to measure in true RMS mode) to ensure that the output voltages are correct. Unusual waveforms on an oscilloscope may indicate that thermal limits are being reached (normally due to an overloaded synchro), and waveforms that drop to zero may indicate that the overvoltage protection was breached and the board has shut down.

Overcurrent messages appear on the serial console and print as follows:

```
Ch0 OC: disabling ist=<...>
Ch1 OC: disabling ist=<...>
```

Alternatively, overcurrent conditions can be monitored in user applications written with the low-level API; the `DqCmdReadStatus()` API returns as STS_POST_OVERCURRENT in the POST word of the board status in the low level framework. Refer to `SampleGetStatus.c` for a programming example of reading the status. (See Section 4.3 for the location of sample programs.)

Chapter 3 Programming with the High-level API

This chapter provides the following information about using the UeiDaq high-level Framework API to program the DNx-AI-255:

- About the High-level Framework (Section 3.1)
- Creating a Session (Section 3.2)
- Configuring the Resource String (Section 3.3)
- Configuring for Input (Section 3.4)
- Configuring for Simulated Synchro/Resolver Output (Section 3.5)
- Configuring the Timing (Section 3.6)
- Reading Data (Section 3.7)
- Writing Data (Section 3.8)
- Cleaning-up the Session (Section 3.9)

3.1 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.

3.2 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, and a session object for output

CUeiSession aiSession;
CUeiSession aoSession;
```

3.3 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 input lines 0,1 on device 1 at IP address 192.168.100.2: “pdna://192.168.100.2/Dev1/Ai0,1”

3.4 Configuring for Input

The AI-255 can be configured for a synchro or resolver input.

Use the method **CreateSynchroResolverChannel()** to program the input channels and parameters associated with each channel.

The following call configures the analog input channels of an AI-255 set as device 1:

```
// Configure session synchro/resolver input

aiSession.CreateSynchroResolverChannel(
                                    "pdna://192.168.100.2/Dev1/Ai0,1",
                                     UeiSynchroMode,
                                     3.0,
                                     5000.0,
                                     false);
```

It configures the following parameters:

- **Sensor Mode**: the type of sensor (synchro or resolver) connected to the input channel.
- **Excitation Voltage**: the amplitude of the excitation sine wave in volts RMS.
- **Excitation Frequency**: the frequency of the excitation sine wave.
- **External Excitation**: specifies whether you wish to provide external excitation or use the excitation provided by the AI-255.

If you want to use different parameters for each channel, you can call **CreateSynchroResolverChannel()** multiple times with a different set of channels (0 or 1) in the resource string.

Note that the external excitation amplitude value that comes back from firmware is a peak-to-peak voltage that is converted to an RMS value by the framework on the assumption that it is a sinusoidal excitation signal. However, position transducers may use a square wave or a pulse for excitation. As a result, the amplitude for these signals will appear to be low, and only serve to verify the existence of a signal. When using the framework, the actual RMS or peak-to-peak amplitude of the excitation signal should be measured using an oscilloscope to ensure correctness.

3.5 Configuring for Simulated Synchro/ Resolver Output

The AI-255 can be configured as a synchro or resolver output. When using the AI-255 in Synchro/Resolver Mode, you can also use the AI-255 to simulate a Synchro or a Resolver output.

The following call configures an analog output channel of an AI-255 set as device 1:

```
// Configure session for synchro/resolver output

aoSession.CreateSimulatedSynchroResolverChannel(
                                  "pdna://192.168.100.2/Dev1/AO0",
                                   UeiResolverMode,
                                   3.0,
                                   5000.0,
                                   false);
```

It configures the following parameters:

- **Sensor Mode**: the type of sensor (synchro or resolver) to be simulated.
- **Excitation Voltage**: the amplitude of the excitation sine wave in volts RMS.
- **Excitation Frequency**: the frequency of the excitation sine wave.
- **External Excitation**: specifies whether you wish to provide external excitation or use the excitation provided by the AI-255.

3.6 Configuring the Timing

You can configure the AI-255 to run in simple mode (point by point) or DMAP mode.

In simple mode, the delay between samples is determined by software on the host computer. In DMAP mode, the delay between samples is determined by the AI-255 on-board clock and data is transferred one scan at a time between PowerDNA and the host PC.

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) for AI

aiSession.ConfigureTimingForSimpleIO();

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

aoSession.ConfigureTimingForSimpleIO();
```

3.7 Reading Data

Reading data is done using *reader* object(s). The following sample code shows how to create a scaled reader object and read samples.

```
// create a reader and link it to the analog-input session's stream
CUeiAnalogScaledReader aiReader(aiSession.GetDataStream());
// the buffer must be big enough to contain one value per channel
double data[2];
// read one scan, where the buffer will contain one value per channel
aiReader.ReadSingleScan(data);
```

3.8 Writing Data

Writing data is done using a *writer* object.

The following sample shows how to create a scaled writer and write samples. The AI-255 simulates angle positions entered in radians.

```
// create a writer and link it to the session's analog-output stream
CUeiAnalogScaledWriter aoWriter(aoSession.GetDataStream());
// to write a value, the buffer must contain one value per channel
double data[2] = { 1.0, 2.0 };
// write one scan, where the buffer contains one value per channel
aoWriter.WriteSingleScan(data);
```

3.9 Cleaning-up the Session

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

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

Chapter 4 Programming with the Low-level API

This chapter provides the following information about programming the AI-255 using the low-level API:

- About the Low-level API (Section 4.1)
- Low-level Functions (Section 4.2)
- Low-level Programming Techniques (Section 4.3)

NOTE: This chapter contains descriptions about the various operating modes and wiring connections of the AI-255, and descriptions of the low-level functions that may be used in programming this module. These functions can also be used by the AI-256. Note that the AI-255 provides lower frequencies and current than the AI-256.

4.1 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 3**). The Framework extends the low-level API with additional functionality that makes programming easier and faster, and additionally the Framework supports a variety of programming languages and the use of scientific software packages such as LabVIEW and Matlab.

For additional information regarding low-level programming, refer to the "PowerDNA API Reference Manual" located in the following directories:

- On Linux systems:
  <PowerDNA-x.y.z>/docs
- On Windows systems:
  Start » All Programs » UEI » PowerDNA » Documentation

4.2 Low-level Functions

Table 4-1 provides a summary of AI-255-specific functions. All low-level functions are described in detail in the PowerDNA API Reference Manual.

Table 4-1. Summary of Low-level API Functions for DNx-AI-255

Function	Description
`DqAdv255SetMode`	Sets synchro or resolver operating modes per channel
`DqAdv255SetExt`	Set up extra (additional) parameters
`DqAdv255SetExcitation`	Sets excitation frequency and amplitude in internal exitation mode
`DqAdv255GetWFMeasurements`	Returns the measured parameters of waveform on selected input(s)

Table 4-1. Summary of Low-level API Functions for DNx-AI-255 (Cont.)

Function	Description
`DqAdv255MeasureWF`	Simple form of DqAdv255GetWFMeasurements
`DqAdv255Enable`	Enables/disables operation on specified channels
`DqAdv255Read`	Reads the calculated angle or special data for selected channels (point-to-point mode)
`DqAdv255Write`	Writes a simulated position of a synchro or resolver or special data (point-to-point mode)
`DqAdv255ConvertSim`	Converts angle to raw data representation for gain and phase control
`DqAdv255WriteBin`	Writes an angle or special data for selected channels

4.3 Low-level Programming Techniques

Application developers are encouraged to explore existing source code examples when first programming the AI-255. Sample code provided with the installation is self-documented and serves as a good starting point.

Code examples are located in the following directories:

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

Sample code for the different 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, SampleDMap255 contains sample code for running the an AI-255 using DMap data acquisition mode. Note that immediate mode samples are named Sample<I/O board name>, (i.e., Sample255).

4.3.1 Data Collection Modes

The AI-255 supports the following acquisition modes:

- Immediate (point-to-point): Designed to provide easy access to a single I/O board at a non-deterministic pace. Acquires a single data point per channel. Runs at a maximum of 100 Hz.
- rtDMAP: Designed for closed-loop (control) applications. Users set up a “map” of I/O boards and channels from which to acquire data. Input data is stored in I/O board FIFOs at a rate paced by its hardware clock; a single API call (refresh) paced by the user application causes data to be collected directly from I/O board FIFOs and packed for delivery to the user application.
  - rtDMAP collects only 1 data sample per channel

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

4.3.2 Configuring Excitation

The DNx-AI-255 can be configured to use internal or external excitation. Before configuring the mode of a channel, users should configure the excitation frequency and level for internal excitation or, if needed, measure the excitation frequency and level for external excitation. These parameters will be needed when setting the mode.

4.3.2.1 Reading External Excitation

If you are supplying external excitation and do not know the excitation frequency and level when writing your application, you can measure excitation parameters using the `DqAdv255GetWFMeasurements` API, which returns a structure that holds the acquired waveform data parameters.

NOTE: This function can take up to 1 second to return; therefore, timeouts programmed in `DqOpenIOM()` should be extended.

The following is an example of reading external excitation:

```
// CHANNELS is 2 for the AI-255
// WFMEASURE_255 & WFPRM_255 are structures for waveform setup & data
for(i=0; i<CHANNELS; i++) {

    WFMEASURE_255 param;
    WFPRM_255 chan_m[DQ_AI255_ADCS]; //A/D per channel: A,B,C & D

    // Configure channel list
    // Set gain = gain of 1
    cl[i] = i | DQ_LNCL_GAIN(DQ_AI255_GAIN_1);  //UEI macro & #define

    //program gain for excitation measurement
    param.changain = cl[i];

    // get waveform frequency and amplitude used to power the sensor
    DqAdv255GetWFMeasurements(
                hd,         // handle for the IOM, set in DqOpenIOM()
                DEVN,       // AI-255 board position in the chassis
                0,          // Reserved, set to 0
                &param,     // structure of type pWFMEASURE_255
                            //   used to store waveform parameters,
                            //   specifically the changain parameter
                            //   must be set to the channel list, cl
                chan_m);    // pointer to structure of type pWFPRM_255,
                            //   used to store external excitation
                            //   measurements
                            //   chan_m.freq: the measured exc frequency
                            //   chan_m.ampl: the measured Vpp level
}
```

Macro, structure and #define descriptions

For more information, refer to the PowerDNA API Reference Manual or refer to the powerdna.h header file, which can be found in the following directories:

- On Windows systems:

```
<UEI installation directory>/PowerDNA/SDK/includes
```

- On Linux systems:

```
<UEI installation directory>/sdk/DAQLIB
```

4.3.2.2 Setting Internal Excitation

To set up internal excitation, use the `DqAdv255SetExcitation` API.

The following is an example of setting up the `DqAdv255SetExcitation` API for generating internal excitation:

```
for(i=0; i<CHANNELS; i++) {
    // Configure channel list
    // Set gain = gain of 1
    cl[i] = i | DQ_LNCL_GAIN(DQ_AI255_GAIN_1);  //UEI macro & #define

    // For internal excitation modes, configure the
    // amplitude and frequency of the sine waveform
    // used to power the sensor
    // convert RMS voltage to peak-to-peak voltage
    exc_rates[i] = INT_EXC_RATE;       // From 50 Hz to 4 kHz
    exc_rms_amplitudes[i] = RMS_AMPL;  // Up to 28 Vrms
    exc_peakpeak_amplitudes[i] = exc_rms_amplitudes[i]*(2*DQ_AI254_AMP2RMS);


    DqAdv255SetExcitation(
          hd,               // handle for the IOM, set in DqOpenIOM()
          DEVN,             // AI-255 board position in the chassis
          DQ_LNCL_GETCHAN(cl[i]), // get channel # programmed in cl
          DQ_AI255_ENABLE_EXC_D,  // set which output to use for
                                  //   excitation, D must be used
                                  //   for simulation devices
          exc_rates[i],
          exc_peakpeak_amplitudes[i]; //peak-to-peak excitation voltage

}
```

4.3.3 Configuring the Operational Mode

The `DqAdv255SetMode() API` configures a DNx-AI-255 channel as a Synchro or Resolver, as an input or output, or to use internal or external excitation, etc.

Since the `DqAdv255SetMode` API is passed an expected excitation frequency (`exc_freq`) and an excitation voltage level (`se_level`), before calling `DqAdv255SetMode()`, call an API to get or set excitation waveform parameters. See Section 4.3.2 for more information about programming excitation.

The `DqAdv255SetMode` API can then be called to set the mode:

```
DqAdv255SetMode(hd,         // handle for the IOM
                DEVN,       // AI-255 board position in the chassis
                channel,    // channel to apply mode setting to
                mode,       // listed below in Table 4-2
                0,          // Reserved, set to 0
                exc_freq,   // expected excitation frequency
                se_level);  // Excitation level (Vpp)
```

The AI-255 can be programmed as synchro or resolver modes with internal or external excitation.

Synchro modes can additionally be configured as z-grounded, allowing the S3 stator to be grounded.

To set the mode of a channel, pass one of the following as the `mode parameter`:

Mode	Description
DQ_AI255_MODE_SI_INT	Synchro input, internal excitation
DQ_AI255_MODE_RI_INT	Resolver input, internal excitation
DQ_AI255_MODE_SI_EXT	Synchro input, external excitation
DQ_AI255_MODE_RI_EXT	Resolver input, external excitation
DQ_AI255_MODE_SS_INT	Synchro simulation (output), internal excitation
DQ_AI255_MODE_RS_INT	Resolver simulation (output), internal excitation
DQ_AI255_MODE_SS_EXT	Synchro simulation (output), external excitation
DQ_AI255_MODE_RS_EXT	Resolver simulation (output), external excitation
DQ_AI255_MODE_SI_INTZ	Synchro input, internal excitation, Z-grounded
DQ_AI255_MODE_SI_EXTZ	Synchro input, external excitation, Z-grounded
DQ_AI255_MODE_SS_INTZ	Synchro simulation (output), internal excitation, Z-grounded
DQ_AI255_MODE_SS_EXTZ	Synchro simulation (output), external excitation, Z-grounded

Table 4-2. AI-255 Modes of Operation

NOTE: More information about modes can be found in `powerdna.h` or the PowerDNA API Reference Manual.

4.3.3.1 Wiring

A synchro has three stator coils S1, S2, S3 connected in a star or delta fashion to the Common. The rotor primary coil (exciter) has wires R1 and R2.

Resolver stator coils are S1-S3 and S2-S4. Rotor coil is R1-R3 (and R2-R4 when two rotor windings are used).

Wiring connections for various modes of operation are provided in Table 2-4 on page 23.

Refer to Appendix B for connection diagrams.

4.3.4 Enabling AI-255 Channels

After the excitation and modes of operation are set up, the DNx-AI-255 channels can be enabled with the `DqAdv255Enable()` API.

```
// Enable channels,
// For example, CHANNELS is set to 2 (the maximum number of
// channels on the device)
// and cl is a configuration array of length CHANNELS, (i.e. cl[0]
// holds the gain setting for channel 0, cl[1] holds the gain setting
// for channel 1)

DqAdv255Enable( hd,          //handle for the IOM
                DEVN,        // AI-255 board position in the chassis
                0,           // Reserved, set to 0
                1,           // 1 = TRUE (enabled)
                CHANNELS,    // CHANNELS = max # of channels / board
                cl);         // channel list, encoded with gain settings
```

4.3.5 Reading Synchro or Resolver Inputs (Immediate Mode)

After the channels are enabled, you can measure the stator voltages on AI-255 channels configured as inputs (not simulated outputs).

The `DqAdv255Read` API is used to acquire stator readings in Immediate (point-to-point) data acquisition mode. To review UEI data acquisition modes, refer to Section 4.3.1 on page 33 for descriptions, and refer to example code for API used in other modes.

```
DqAdv255Read(   hd,          // handle for the IOM
                DEVN,        // AI-255 board position in the chassis
                CHANNELS,    // CHANNELS = max # of channels / board
                cl,          // channel list
                NULL,        //
                angles);     // converted data
```

By default, `DqAdv255Read` returns the calculated rotor angle (`angles`).

4.3.6 Writing Synchro or Resolver Simulated Outputs (Immediate Mode)

After the channels are enabled, you can output simulated stator voltages on AI-255 channels configured as simulated outputs.

The `DqAdv255Write` API is used in Immediate data acquisition mode to convert an angle parameter to stator output waveforms and output those waveforms. To review UEI data acquisition modes, refer to Section 4.3.1 on page 33 for descriptions, and refer to example code for API used in other modes.

```
double angles = angleInDegrees * _pi_/180.0;

DqAdv255Write(  hd,         // handle for the IOM
                DEVN,       // AI-255 board position in the chassis
                CHANNELS,   // CHANNELS = max # of channels / board
                cl,         // channel list
                exc_peakpeak_amplitudes,
                            // maximum voltage (Vpp) of the simulated
                            // stator voltage
                angles);    // output position in radians
```

Note that if you review stator waveforms on a scope, you are looking at line-to-ground waveforms. For more information about interpreting simulated outputs and to review simulated position calculations and formulas, refer to the `DqAdv255Write` API description in the PowerDNA API Reference Manual.

Appendix A

A.1 Accessories

The following cables and STP boards are available for the AI-255 board.

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.

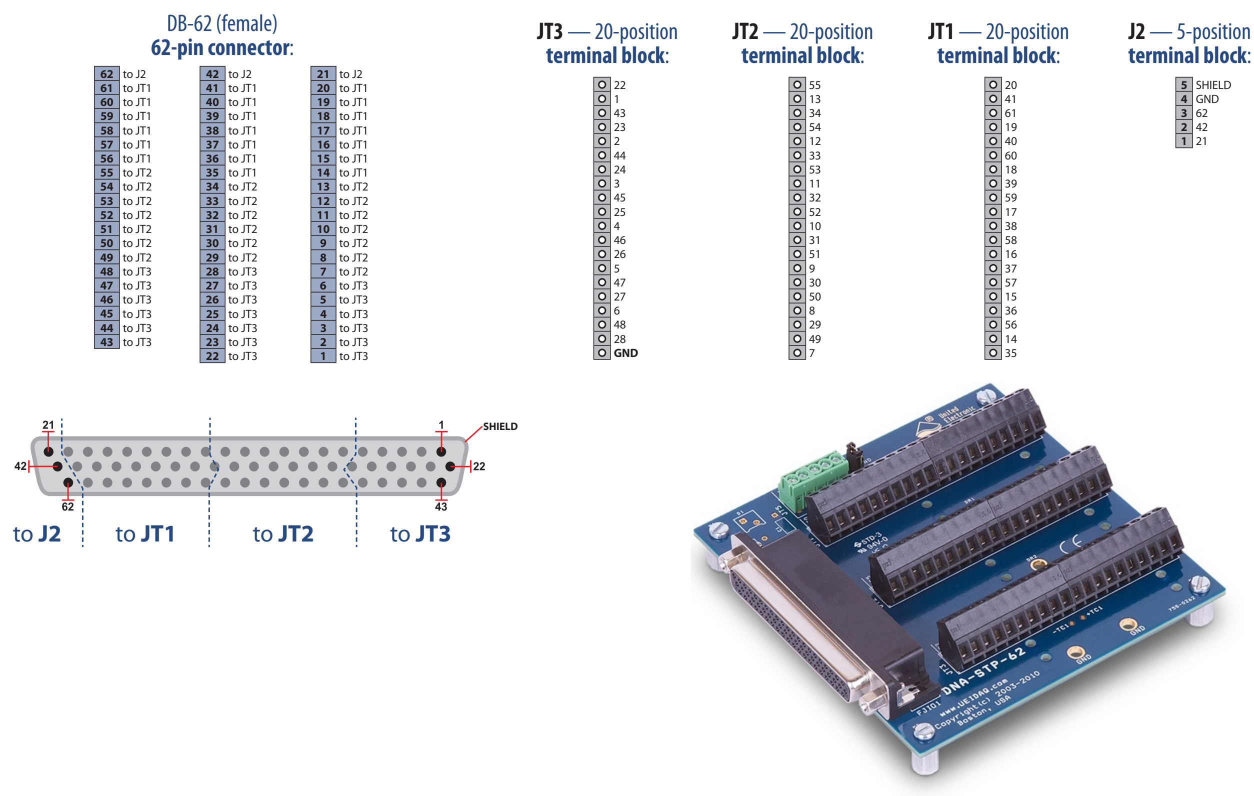


Figure A-1 Pinout and photo of DNA-STP-62 screw terminal panel

Appendix B

Connection Diagrams for Synchro / Resolver Operating and Excitation Modes

This appendix provides connection diagrams for the following AI-255 modes:

- Synchro Input Mode with Internal Excitation (Section B.1)
- Synchro Input Mode with External Excitation (Section B.2)
- Synchro Simulator Output Mode with Internal Excitation (Section B.3)
- Synchro Simulator Output Mode with External Excitation (Section B.4)
- Synchro Simulator Output Mode with External Excitation & Z-grounding (Section B.5)
- Resolver Input Mode with Internal Excitation (Section B.6)
- Resolver Input Mode with External Excitation (Section B.7)
- Resolver Simulator Output Mode with Internal Excitation (Section B.8)
- Resolver Simulator Output Mode with External Excitation (Section B.9)

B.1 Synchro Input Mode with Internal Excitation

Set up AI-255 connections as shown in **Figure B-1** (star synchro) or **Figure B-2** (delta synchro) when a channel is configured for the following:

- Each stator line of the synchro is input to AI-255
- AI-255 generates excitation internally

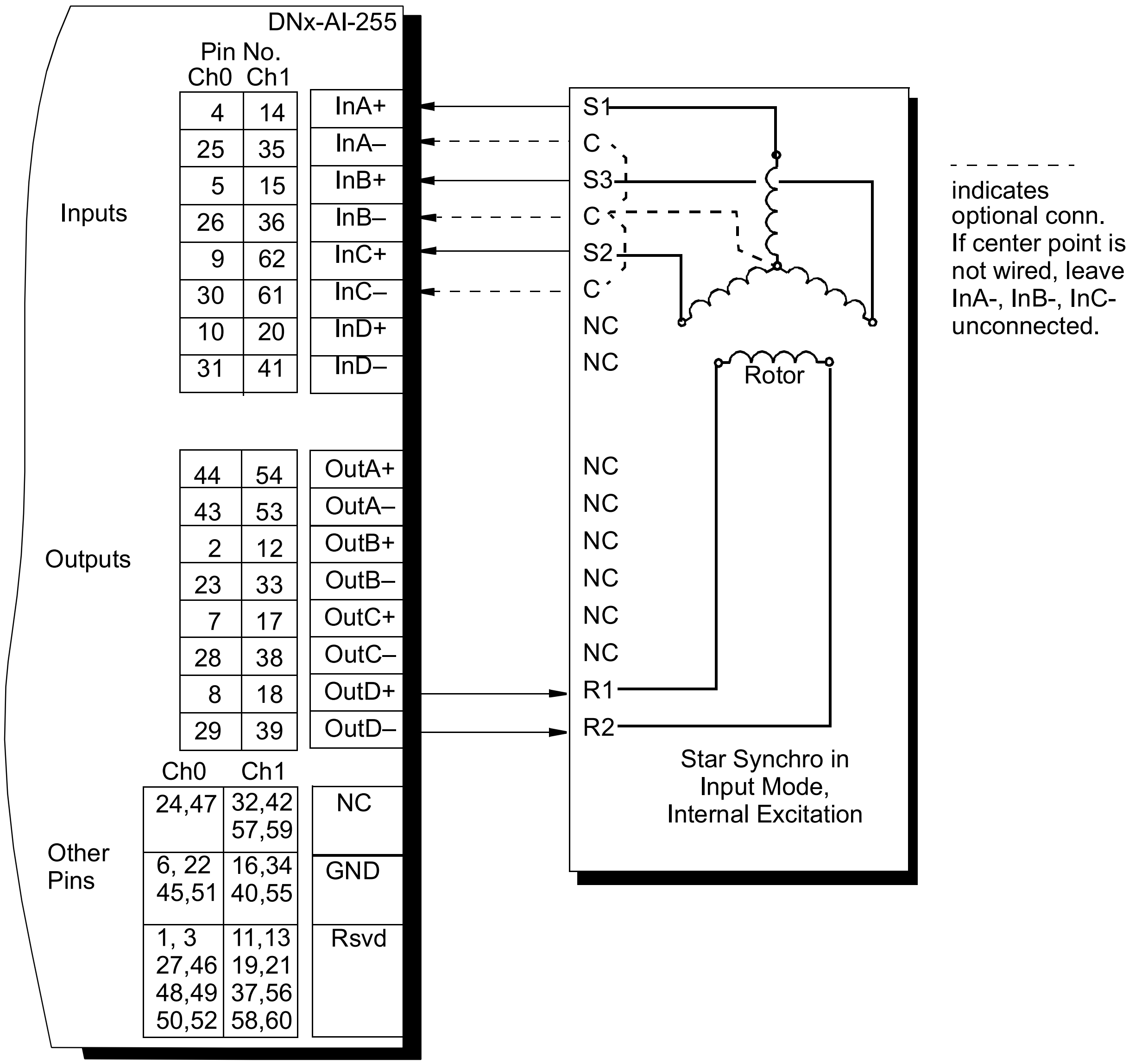


Figure B-1 Star Synchro: Input Mode, Internal Excitation

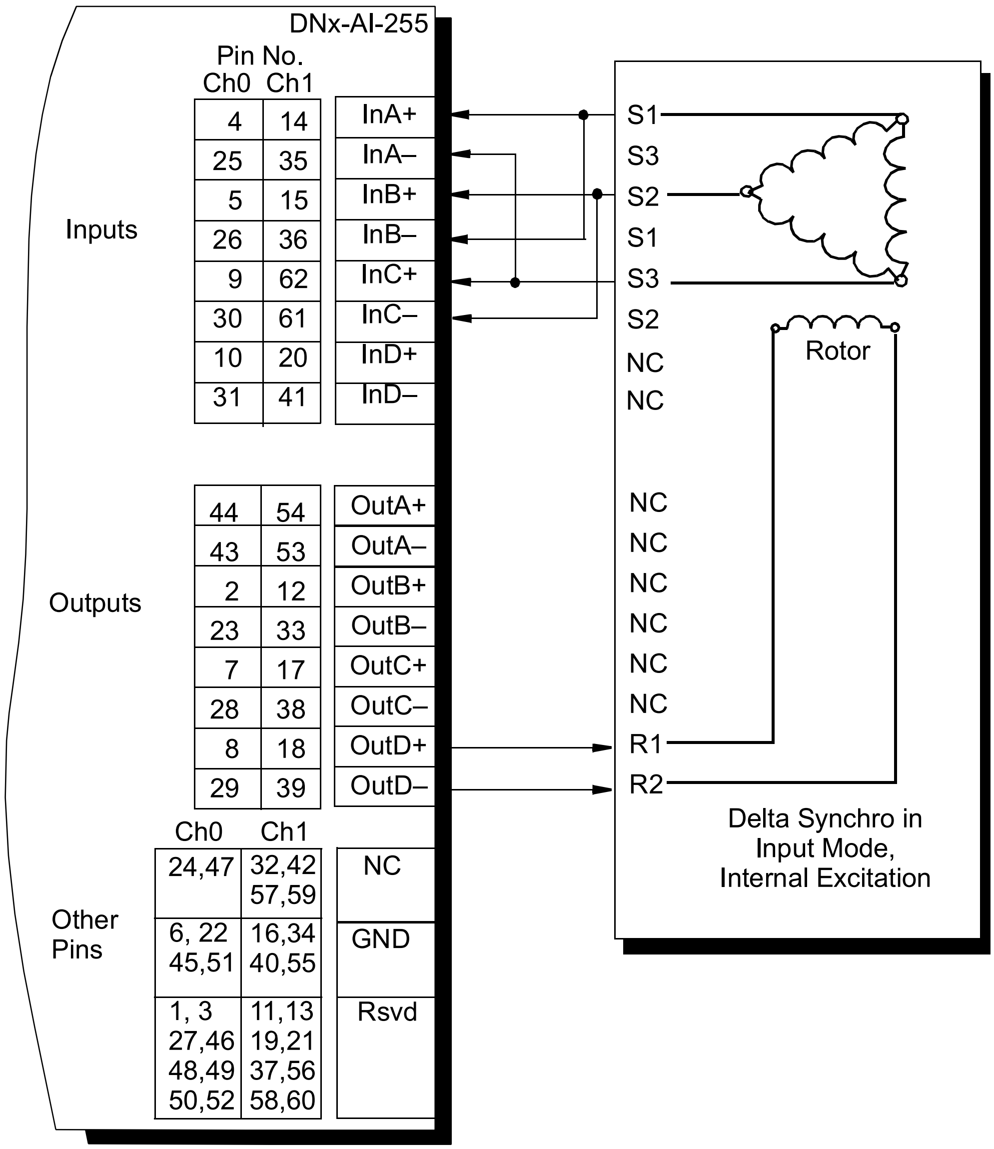


Figure B-2 Delta Synchro: Input Mode, Internal Excitation

B.2 Synchro Input Mode with External Excitation

Set up AI-255 connections as shown in **Figure B-3** (star synchro) or **Figure B-4** (delta synchro) when a channel is configured for the following:

- Each stator line of the synchro is input to AI-255
- AI-255 uses an externally generated excitation

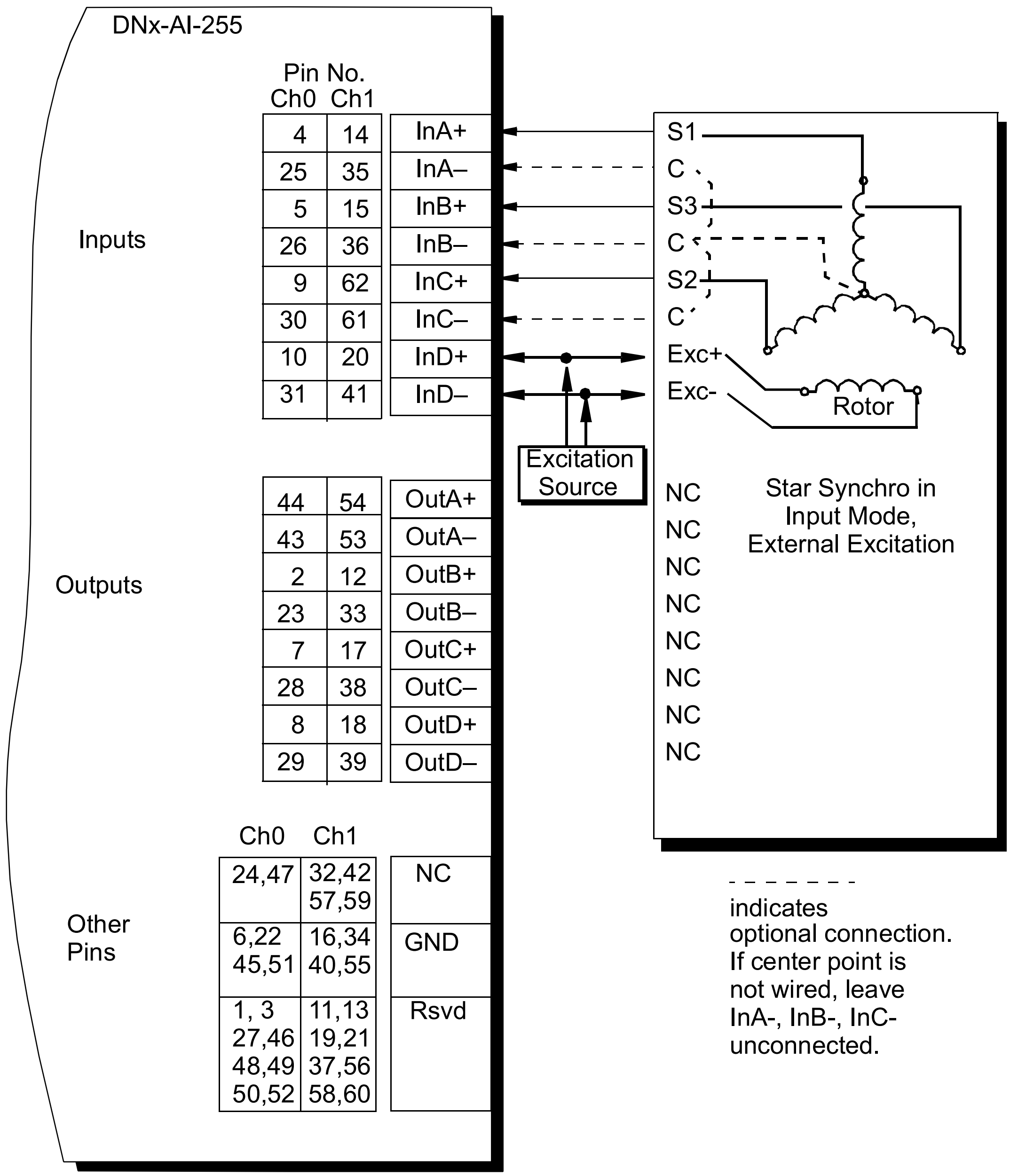


Figure B-3 Star Synchro: Input Mode, External Excitation

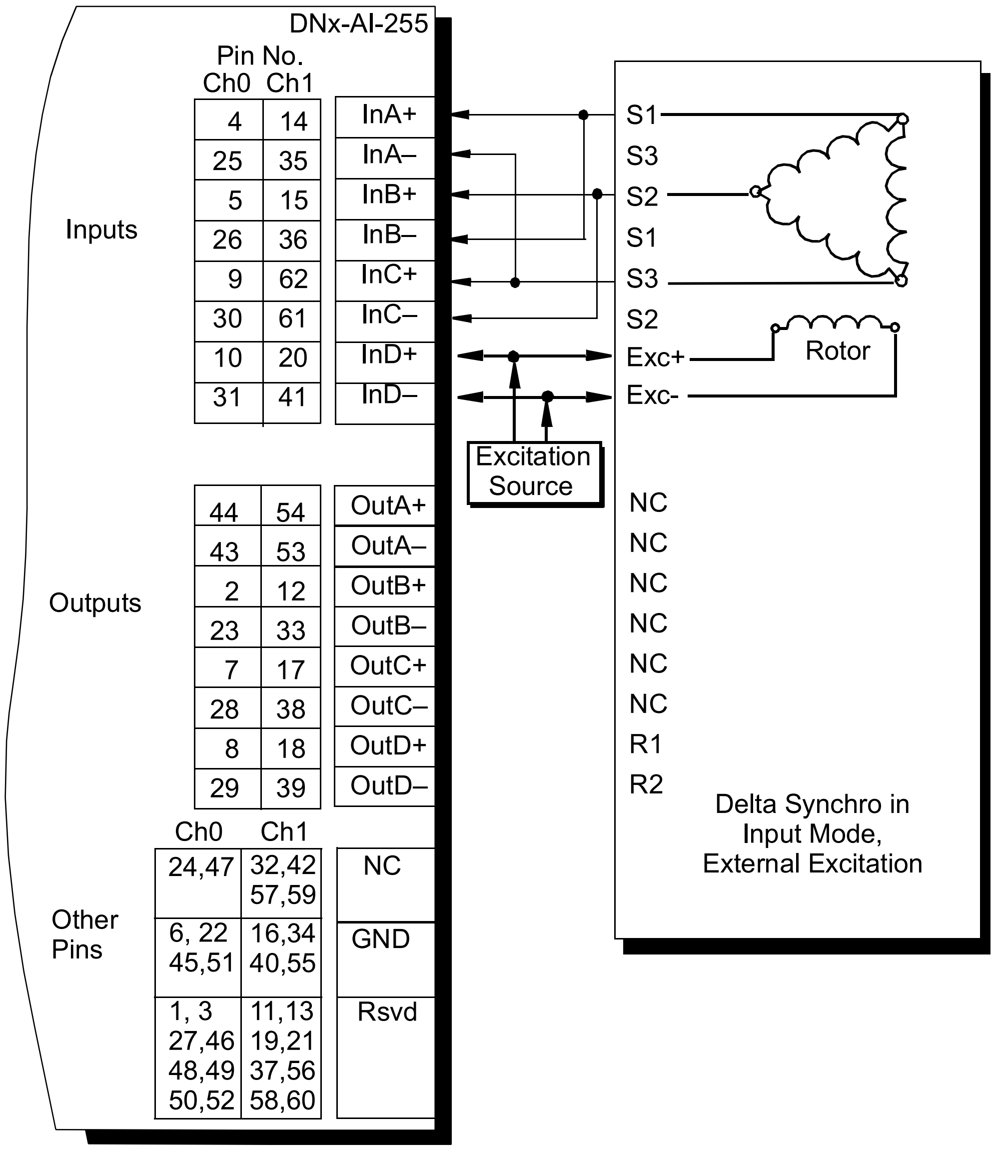


Figure B-4 Delta Synchro: Input Mode, External Excitation

B.3 Synchro Simulator Output Mode with Internal Excitation

Set up AI-255 connections as shown in **Figure B-5** (star synchro) or **Figure B-6** (delta synchro) when a channel is configured for the following:

- AI-255 simulates a synchro and internally generates stator line waveforms
- AI-255 generates excitation internally

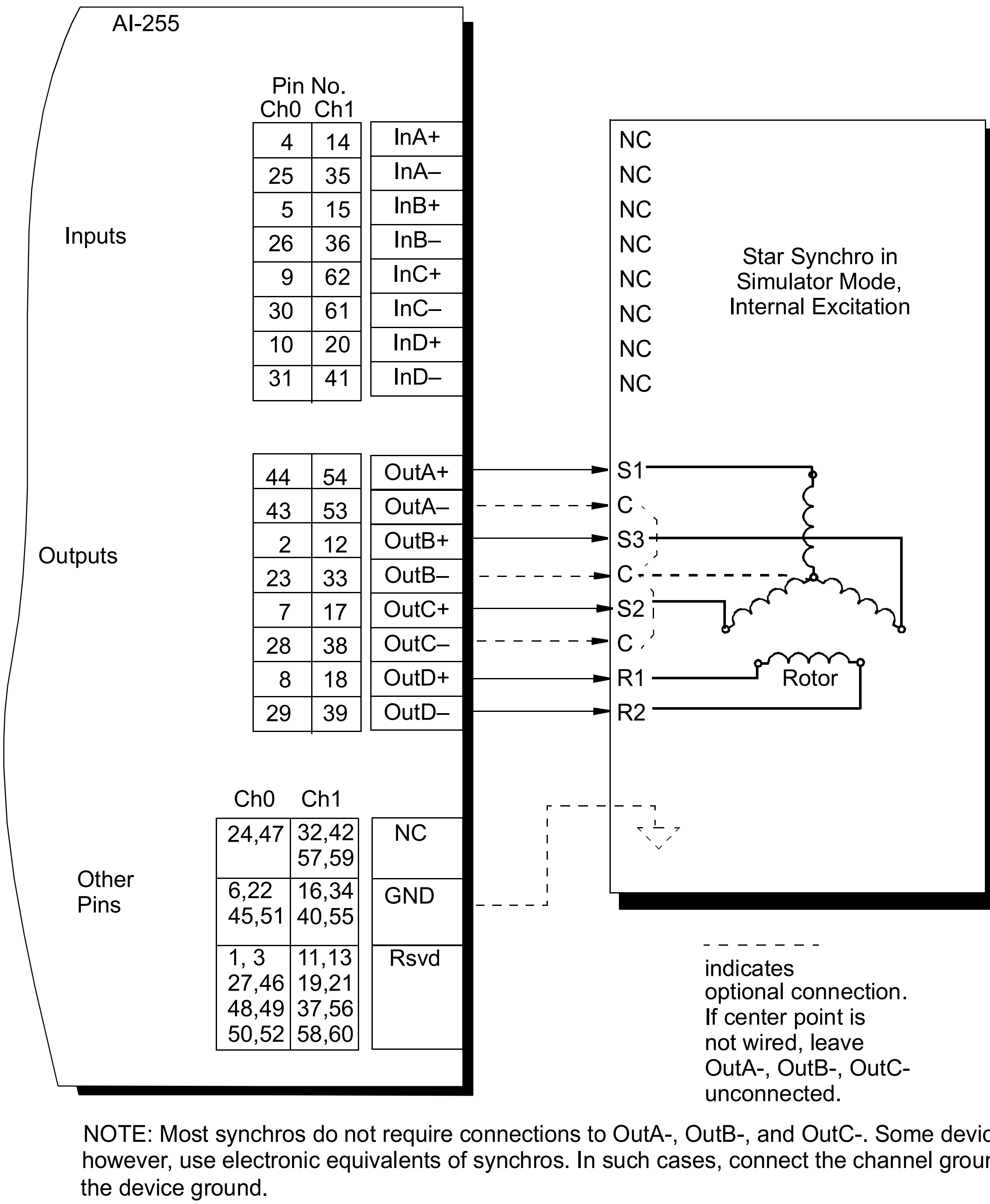


Figure B-5 Star Synchro: Simulator Mode, Internal Excitation

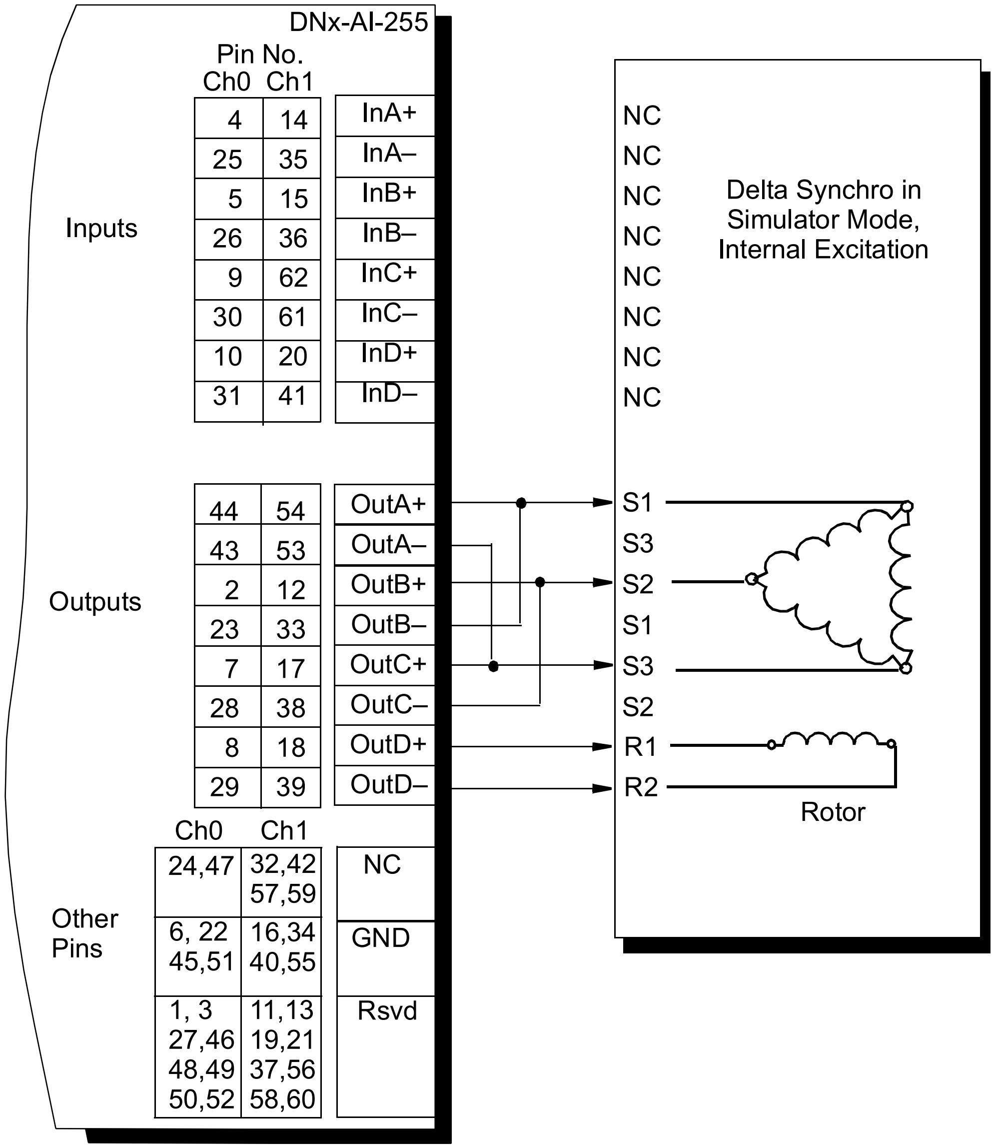


Figure B-6 Delta Synchro: Simulator Mode, Internal Excitation

B.4 Synchro Simulator Output Mode with External Excitation

Set up AI-255 connections as shown in **Figure B-7** (star synchro) or **Figure B-8** (delta synchro) when a channel is configured for the following:

- AI-255 simulates a synchro and internally generates stator line waveforms
- AI-255 uses an externally generated excitation source

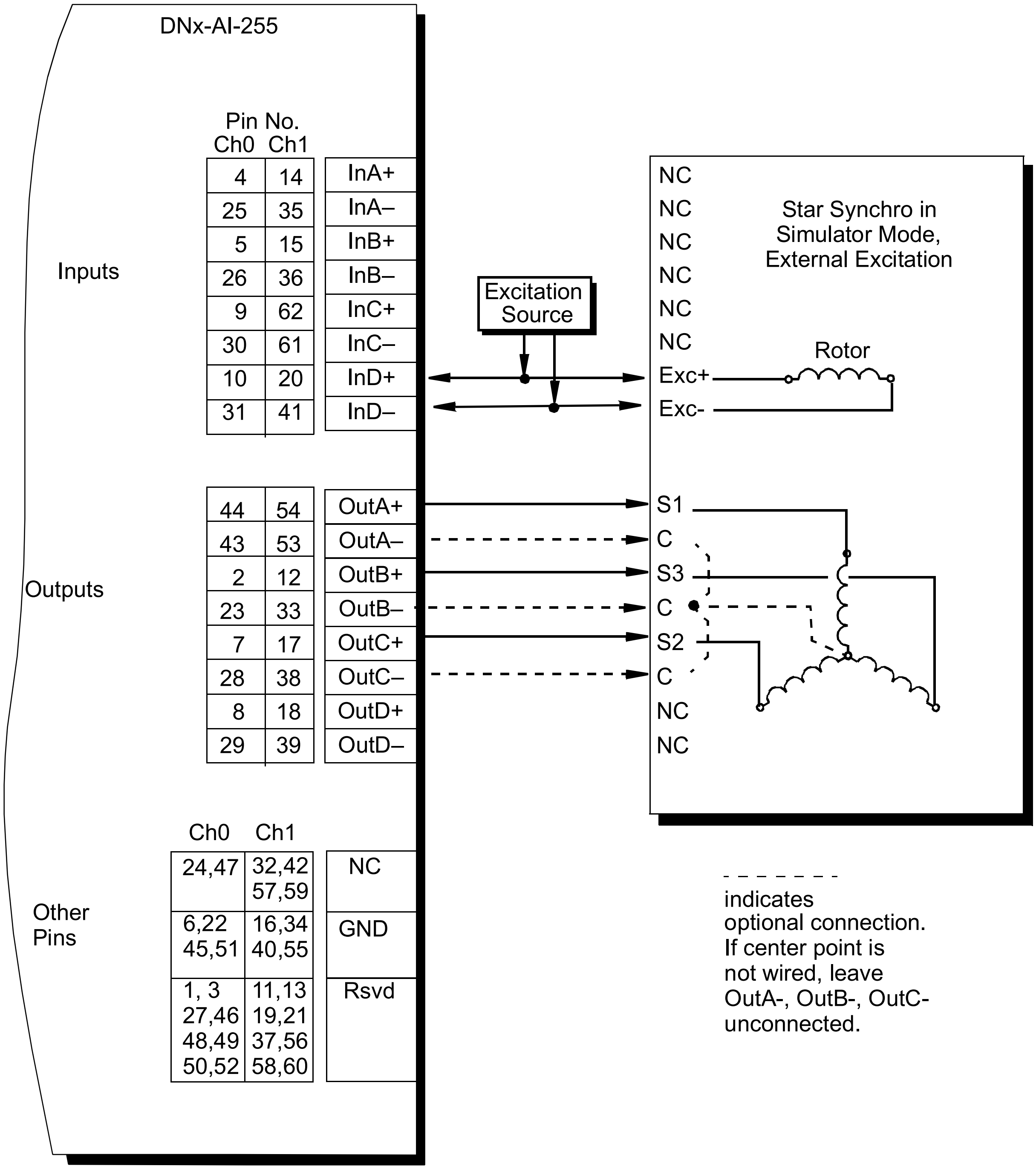


Figure B-7 Star Synchro: Simulator Mode, External Excitation

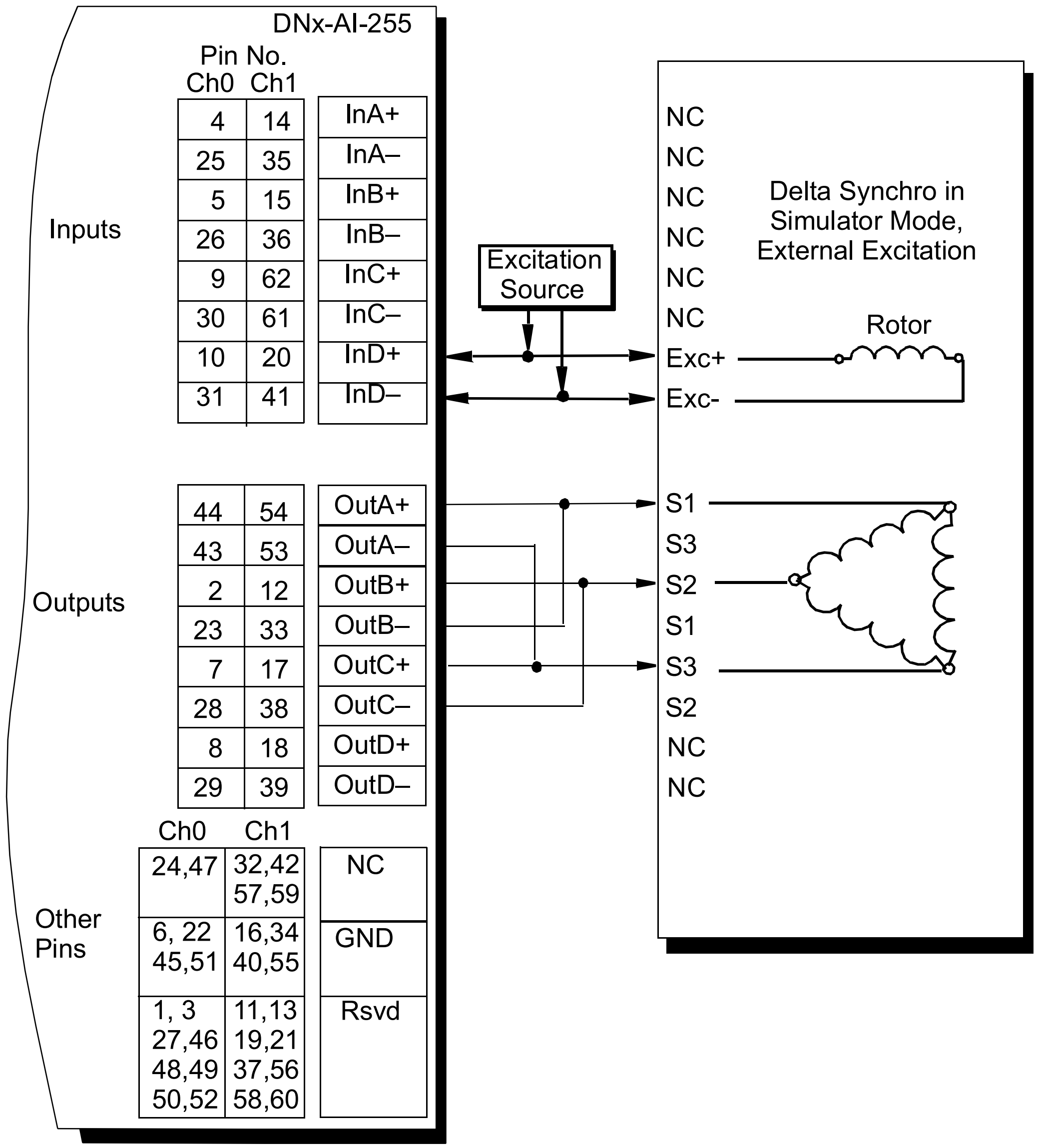


Figure B-8 Delta Synchro: Simulator Mode, External Excitation

B.5 Synchro Simulator Output Mode with External Excitation & Z-grounding

Set up AI-255 connections as shown in **Figure B-9** (star synchro) when a channel is configured for the following:

- AI-255 simulates a synchro and internally generates stator line waveforms**
- **AI-255 is configured for z-grounding mode and produces waveforms on S1 and S2 accordingly: OutC is open and S3 on mating device is grounded
- AI-255 uses an externally generated excitation source

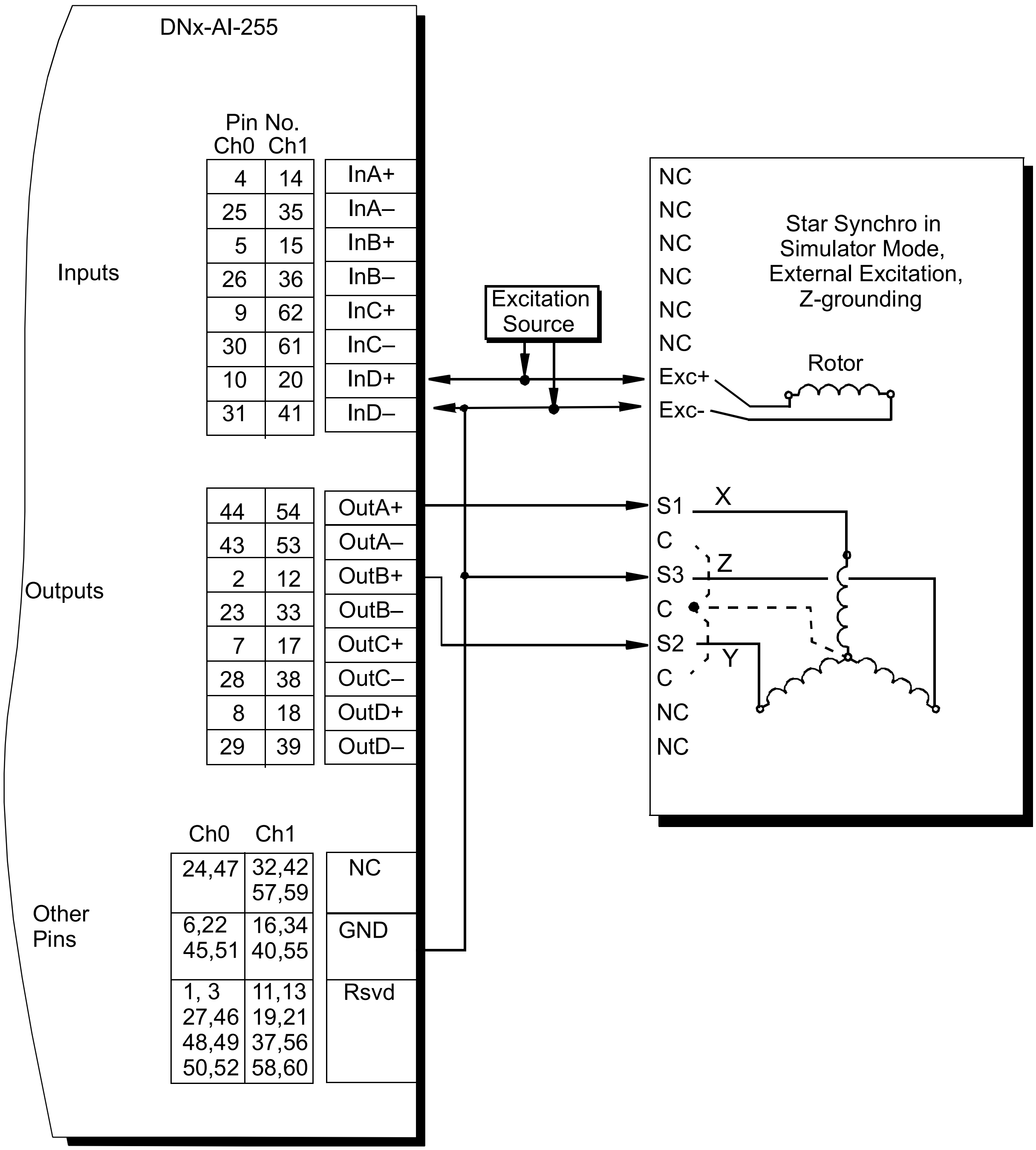


Figure B-9 Star Synchro: Simulator Mode, External Excitation, Z-grounding

B.6 Resolver Input Mode with Internal Excitation

Set up AI-255 connections as shown in **Figure B-10** when a channel is configured for the following:

- Each stator line of the resolver is input to AI-255
- AI-255 generates excitation internally

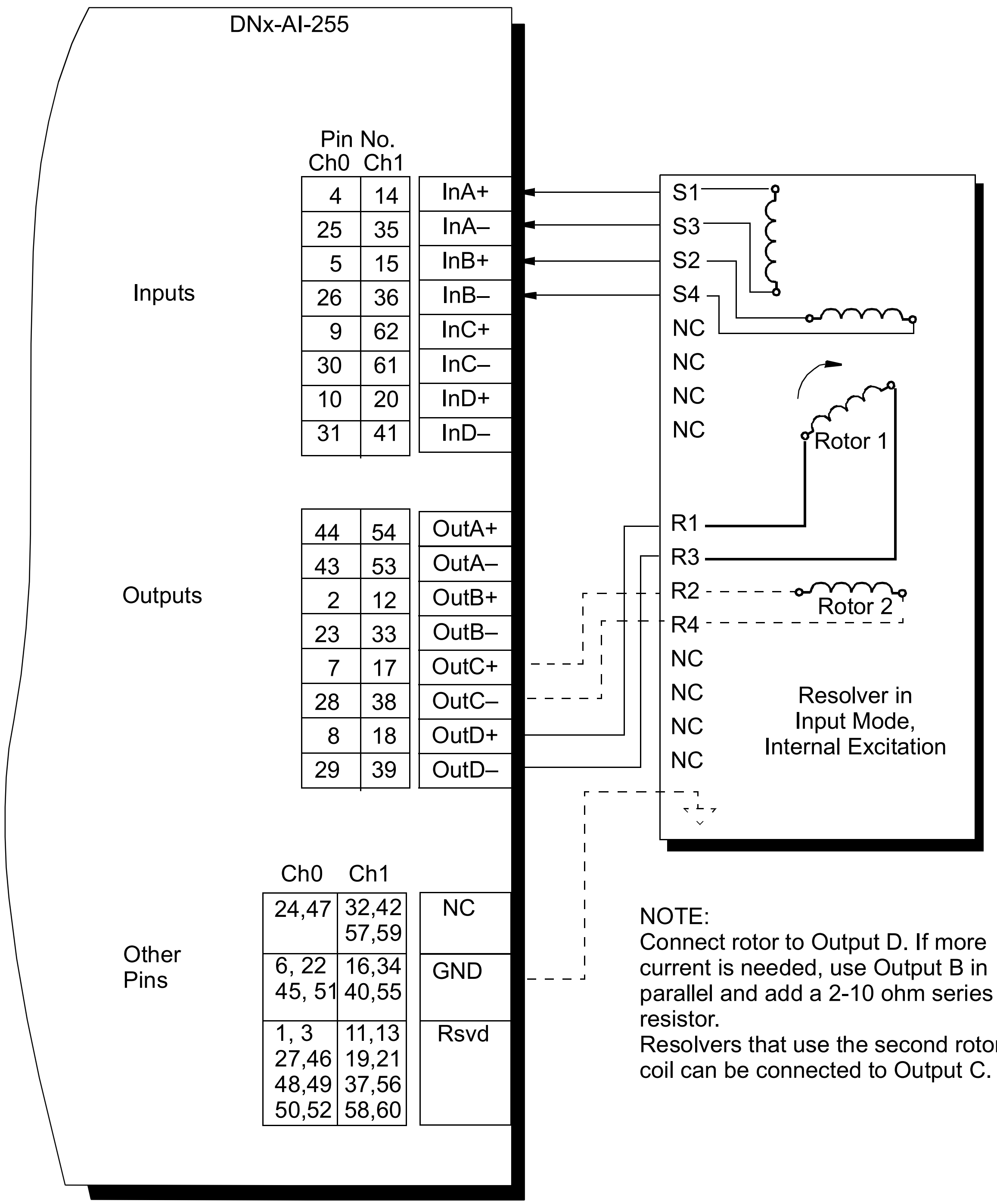


Figure B-10 Resolver: Input Mode, Internal Excitation

B.7 Resolver Input Mode with External Excitation

Set up AI-255 connections as shown in **Figure B-11** when a channel is configured for the following:

- Each stator line of the resolver is input to AI-255
- AI-255 uses an externally generated excitation source

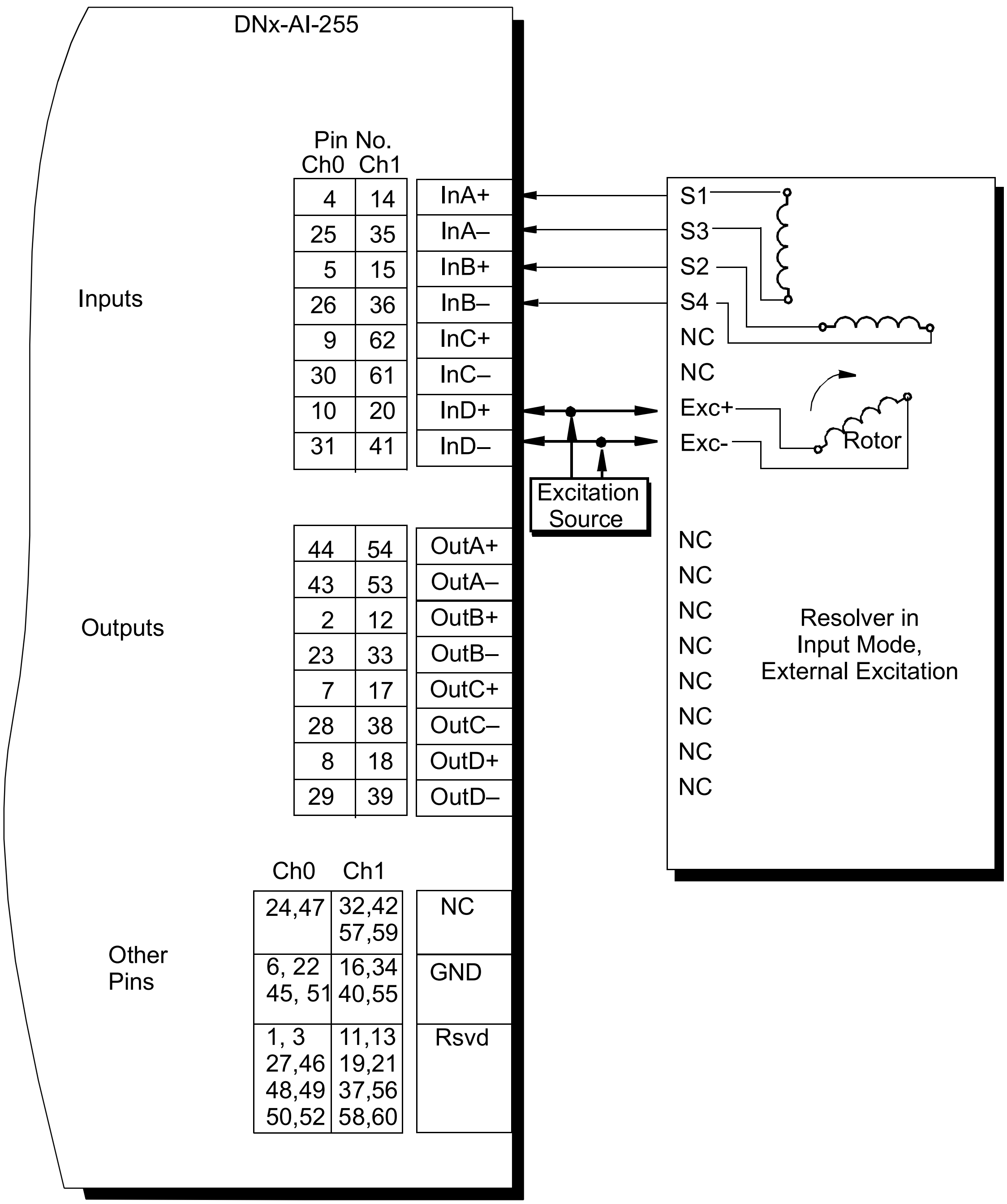


Figure B-11 Resolver: Input Mode, External Excitation

B.8 Resolver Simulator Output Mode with Internal Excitation

Set up AI-255 connections as shown in **Figure B-12** when a channel is configured for the following:

- AI-255 simulates a resolver and internally generates stator line waveforms
- AI-255 generates excitation internally

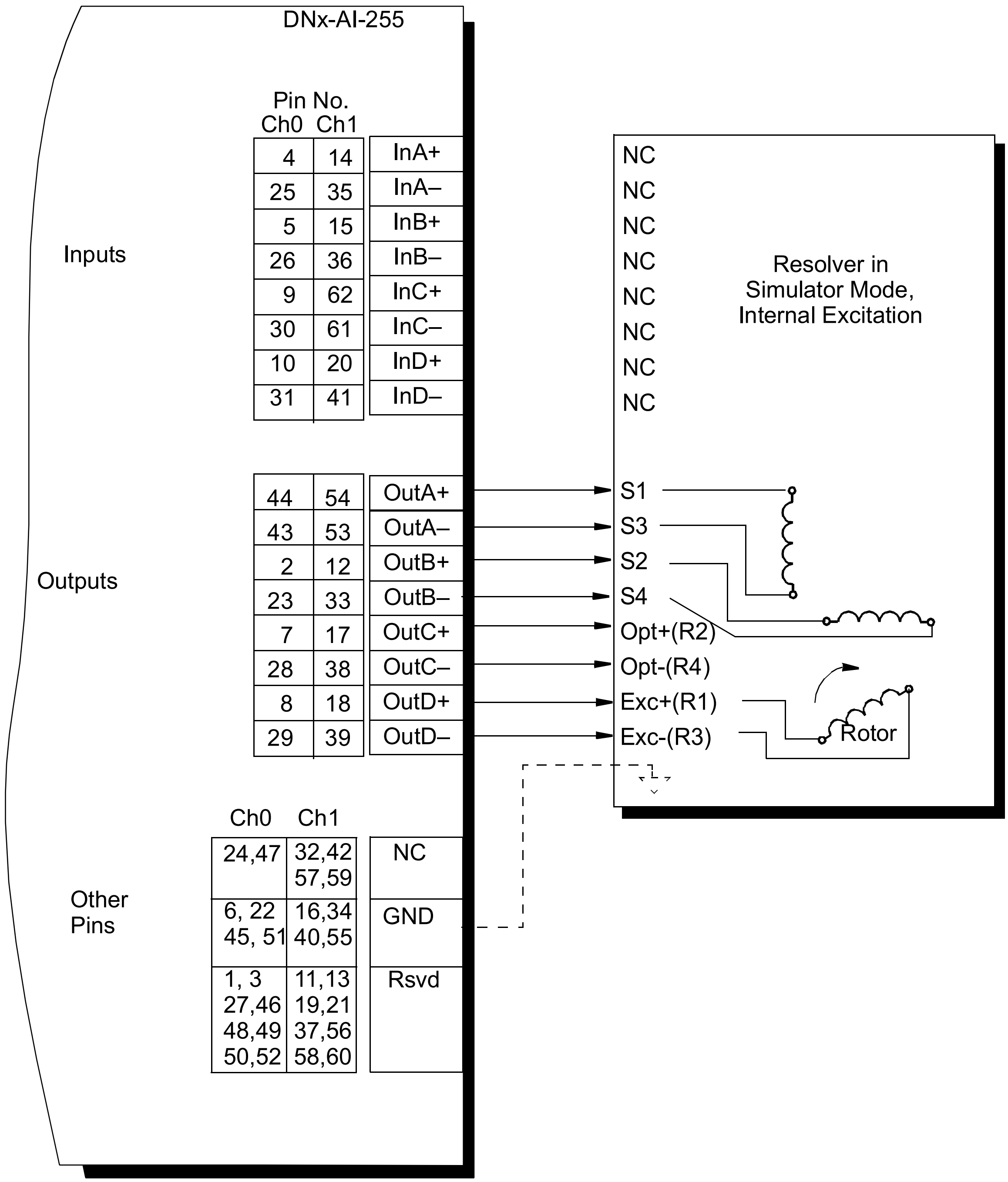


Figure B-12 Resolver: Simulator Mode, Internal Excitation

B.9 Resolver Simulator Output Mode with External Excitation

Set up AI-255 connections as shown in **Figure B-13** when a channel is configured for the following:

- AI-255 simulates a resolver and internally generates stator line waveforms
- AI-255 uses an externally generated excitation source

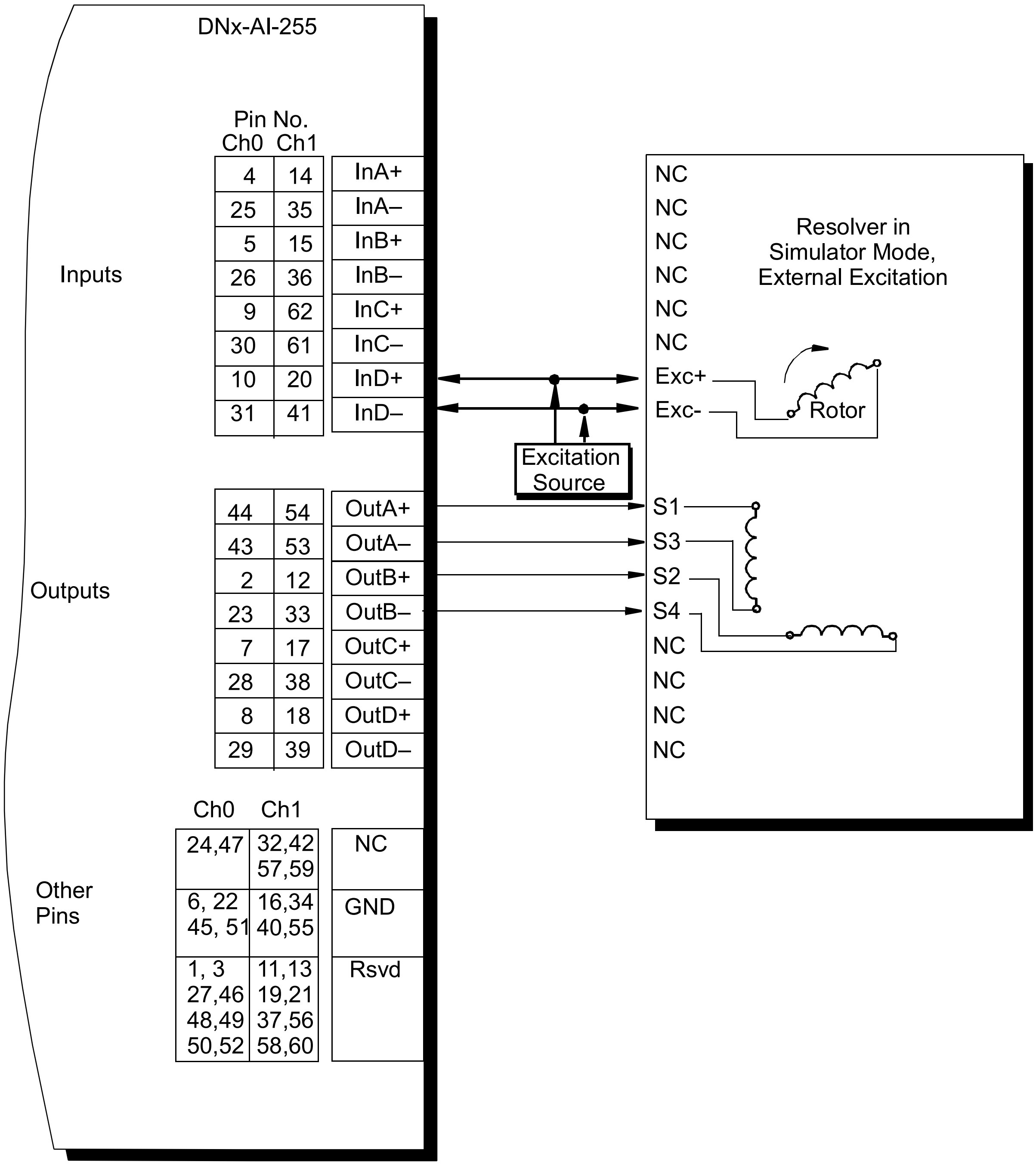


Figure B-13 Resolver: Simulator Mode, External Excitation