AI_Configure

Format

status = `AI_Configure` **(deviceNumber, chan, inputMode, inputRange, polarity, driveAIS)**

Purpose

Informs NI-DAQ of the input mode (single-ended or differential), input range, and input polarity selected for the device. Use this function if you have changed the jumpers affecting the analog input configuration from their factory settings. For devices that have no jumpers for analog input configuration, this function programs the device for the settings you want.

Parameters

Input

Name	Type	Description
deviceNumber	i16	assigned by configuration utility
chan	i16	channel to be configured
inputMode	i16	indicates whether channels are configured for single-ended or differential operation
inputRange	i16	voltage range of the analog input channels
polarity	i16	indicates whether the ADC is configured for unipolar or bipolar operation
driveAIS	i16	indicates whether to drive AISENSE to onboard ground

Parameter Discussion

chan is the analog input channel to be configured, and since the same analog input configuration applies to all of the channels, except for the E Series devices, set **chan** to –1. For the E Series devices, **chan** specifies the channel to be configured. If you want all of the channels to be configured identically, set **chan** to –1.

Range: See Table B-1, *Valid Analog Input Channel Settings*, in Appendix B, *Analog Input Channel, Gain Settings, and Voltage Calculation.*

inputMode indicates whether the analog input channels are configured for single-ended or differential operation.

0: Differential (DIFF) configuration (default).

1: Referenced Single-Ended (RSE) configuration (used when the input signal does not have its own ground reference. The negative input of the instrumentation amplifier is tied to the instrumentation amplifier signal ground to provide one.)

2: Nonreferenced Single-Ended (NRSE) configuration (used when the input signal has its own ground reference. The ground reference for the input signal is connected to AISENSE, which is tied to the negative input of the instrumentation amplifier.)

inputRange is the voltage range of the analog input channels. **polarity** indicates whether the ADC is configured for unipolar or bipolar operation.

0: Bipolar operation (default value).

1: Unipolar operation.

Table 2-1 shows all possible settings for **inputMode**, **inputRange**, and **polarity**. **inputMode** is independent of **inputRange** and **polarity**. In this table, italic text denotes default settings.

Table 2-1. Parameter Settings for `AI_Configure`

Device	Possible Values for inputMode	Analog Input Range			Software Configurable
		inputRange	polarity	Resulting Analog Input Range	
12-bit E Series and 6052E devices	*0*, 1, 2	ignored	unipolar	0 to +10 V	Yes
		ignored	*bipolar*	–5 to +5 V	
16-bit E Series, (except 6052E devices)	*0*, 1, 2	ignored	unipolar	0 to +10 V	Yes
		ignored	*bipolar*	–10 to +10 V	
PCI-6110E, PCI-6111E	0	ignored	*bipolar*	–10 to 10 V	No
Lab-PC+	0, *1*, 2	ignored	unipolar	0 to +10 V	No
		ignored	*bipolar*	–5 to +5 V	
1200 and 1200AI Devices	0, *1*, 2	ignored	unipolar	0 to +10 V	Yes
		ignored	*bipolar*	–5 to +5 V	

Table 2-1. Parameter Settings for `AI_Configure` (Continued)

Device	Possible Values for inputMode	Analog Input Range			Software Configurable
		inputRange	**polarity**	**Resulting Analog Input Range**	
LPM Devices (RSE **inputMode** only)	ignored	5	unipolar	0 to +5 V	No (PC-LPM-16) Yes (PC-LPM-16PnP)
		5	bipolar	–2.5 to +2.5 V	
		10	unipolar	0 to +10 V	
		10	*bipolar*	–5 to +5 V	
516 Devices, DAQCard-500	1	10	*bipolar*	–5 to 5 V	N/A
DAQCard-700	0, *1*	5	bipolar	–2.5 to +2.5 V	Yes
		10	bipolar	–5 to +5 V	
		20	*bipolar*	–10 to +10 V	

Note If a device is software-configurable, the **inputMode**, **inputRange**, and **polarity** parameters are used to program the device for the configuration you want. If a device is not software configurable, this function uses these parameters to inform NI-DAQ of the device configuration, which you must set using hardware jumpers. If your device is software configurable and you have changed the analog input settings through Measurement & Automation Explorer, you do not have to use `AI_Configure`, although it is good practice to do so in case you inadvertently change the configuration file maintained by Measurement & Automation Explorer.

driveAIS is ignored for all other devices. This parameter is present for compatibility reasons only.

Notice that if you have configured any of the input channels in non-reference single-ended (NRSE) mode, this function returns a warning, **inputModeConflict** (18), if you set **driveAIS** to 1. When NI-DAQ reads a channel in NRSE mode, the device uses AISENSE as an input to the negative input of the amplifier, regardless of the **driveAIS** setting. When NI-DAQ reads a channel in differential or reference single-ended (RSE) mode, the device drives AISENSE to onboard ground if **driveAIS** is 1.

Using This Function

When you attach an SC-2040 or SC-2042-RTD to your DAQ device, you must configure channels 0 through 7 for differential mode. When you attach an SC-2043-SG or any SCC accessories to your DAQ device, you must configure these channels for nonreferenced single-ended mode. On the 16-bit E Series devices, the calibration constants used for analog input change depending on the polarity of the analog input channels. NI-DAQ always ensures that the calibration constants in use match the current polarity of the channels.

See the `Calibrate_E_Series` function description for information about calibration constant loading on the E Series devices.

Note The actual loading of calibration constants takes place when you call an `AI`, `DAQ`, or `SCAN` function. On the AT-MIO-16X, the need for reloading the constants depends on the polarity of the channel on which you are doing analog input.

AI_VRead

Format

status = `AI_VRead` **(deviceNumber, chan, gain, voltage)**

Purpose

Reads an analog input channel (initiates an A/D conversion on an analog input channel) and returns the result scaled to a voltage in units of volts.

Parameters

Input

Name	Type	Description
deviceNumber	i16	assigned by configuration utility
chan	i16	analog input channel number
gain	i16	gain setting to be used for the specified channel

Output

Name	Type	Description
voltage	*f64	the measured voltage returned, scaled to units of volts

Parameter Discussion

chan is the analog input channel number.

Range: See Table B-1, *Valid Analog Input Channel Settings*, in Appendix B, *Analog Input Channel, Gain Settings, and Voltage Calculation*.

gain is the gain setting to be used for the specified channel. Refer to Appendix B, *Analog Input Channel, Gain Settings, and Voltage Calculation,* for valid gain settings. If you use an invalid gain, NI-DAQ returns an error. If you call `AI_VRead` for the 516 and LPM devices or DAQCard-500/700, NI-DAQ ignores the gain.

voltage is the floating-point variable in which NI-DAQ returns the measured voltage, scaled to units of volts.

Note C Programmers—**voltage** is a pass-by-address parameter.

Using This Function

AI_VRead addresses the specified analog input channel, changes the input gain to the specified gain setting, and initiates an A/D conversion. AI_VRead waits for the conversion to complete and then scales and returns the result. If the conversion does not complete within a reasonable time, the call to AI_VRead is said to have *timed out* and NI-DAQ returns the **timeOutError** code.

When you use SCXI as a front end for analog input to an MIO or AI device, 1200 Series device, LPM device, or DAQCard-700, it is not advisable to use the AI_VRead function because that function does not take into account the gain of the SCXI module when scaling the data. Use the AI_Read function to get unscaled data, and then call the SCXI_Scale function.

When you have an SC-2040 accessory connected to an E Series device, this function takes both the onboard gains and the gains on SC-2040 into account while scaling the data. When you have an SC-2043-SG accessory connected to your DAQ device, this function takes both the onboard gains and the SC-2043-SG fixed gain of 10 into account while scaling the data.

When you have any SCC accessories connected to an E Series device, this function takes both the onboard gains and the SCC gains into account while scaling the data.

B

Analog Input Channel, Gain Settings, and Voltage Calculation

This appendix lists the valid channel and gain settings for DAQ boards, describes how NI-DAQ calculates voltage, and describes the measurement of offset and gain adjustment.

DAQ Device Analog Input Channel Settings

Table B-1 lists the valid analog input (ADC) channel settings. If you have one or more AMUX-64T boards and an MIO board, see Chapter 10, *AMUX-64T External Multiplexer Device*, in the *DAQ Hardware Overview Guide* for more information.

Table B-1. Valid Analog Input Channel Settings

Device	Settings	
	Single-ended Configuration	**Differential Configuration**
MIO and AI devices (except as noted below)	0–15	0–7
AT-MIO-64E-3	0–63	0–7, 16–23, 32–39, 48–55
Lab and 1200 Series devices	0–7	0, 2, 4, 6
LPM devices	0–15	—
DAQCard-700	0–15	0–7
516 devices, DAQCard-500	0–7	0–3 (516 devices only)
VXI-MIO-64E-1 and VXI-MIO-64XE-10	0–63 and `ND_VXI_SC`	0–7, 16–23, 32–39, 48–55

Table B-1. Valid Analog Input Channel Settings (Continued)

Device	Settings	
	Single-ended Configuration	**Differential Configuration**
DAQPad-MIO-16XE-50	0–15 and `ND_CJ_TEMP`†	0–7 and `ND_CJ_TEMP`†
PCI-6110E PCI-6111E	— —	0–3 0–1
PXI MIO and AI devices	`ND_PXI_SC`	—
PCI-4451 NI 4551	—	0–1
PCI-4452 NI 4552	—	0–3
PCI-4453	0–1	—
PCI-4454	0–3	—

† `ND_CJ_TEMP`, `ND_PXI_SC`, and `ND_VXI_SC` are constants that are defined in the following header files:

- C programmers—`NIDAQCNS.H` (`DATAACQ.H` for LabWindows/CVI)
- BASIC programmers—`NIDAQCNS.INC` (Visual Basic for Windows programmers should refer to the *Programming Language Considerations* section in Chapter 1, *Using the NI-DAQ Functions*, for more information.)
- Pascal programmers—`NIDAQCNS.PAS`

Valid Internal Analog Input Channels

Table B-2 lists the valid internal channels for analog input devices.

Table B-2. Valid Internal Analog Input Channels

Device	Internal Channels

AT-MIO-16XE-10
AT-MIO-16XE-50
NEC-MIO-16XE-50
DAQPad-MIO-16XE-50
	`ND_INT_AI_GND`
	`ND_INT_REF_5V`
	`ND_INT_AO_GND_VS_AI_GND`
	`ND_INT_AO_CH_0`
	`ND_INT_CH_0_VS_REF_5V`
	`ND_INT_AO_CH_1`
	`ND_INT_AO_CH_1_VS_REF_5V`

Table B-2. Valid Internal Analog Input Channels (Continued)

Device	Internal Channels
DAQCard-AI-16E-4 NEC-AI-16E-4	ND_INT_AI_GND ND_INT_REF_5V ND_INT_CM_REF_5V ND_INT_AO_GND_VS_AI_GND
PCI-MIO-16XE-10 PCI-MIO-16XE-50 PXI-6030E PXI-6011E PCI-6031E CPCI-6030E CPCI-6011E VXI-MIO-64XE-10	ND_INT_AI_GND ND_INT_REF_5V ND_INT_AO_GND_VS_AI_GND ND_INT_AO_CH_0 ND_INT_AO_CH_0_VS_REF_5V ND_INT_AO_CH_1 ND_INT_AO_CH_1_VS_REF_5V ND_INT_AO_CH_1_VS_AO_CH_0 ND_INT_DEV_TEMP
PCI-MIO-16E-1 PCI-MIO-16E-4 PXI-6070E PXI-6040E CPCI-6070E CPCI-6040E VXI-MIO-64E-1 DAQPad-6070E PCI-6024E PCI-6025E PXI-6025E PCI-6035E PXI-6035E PCI-6052E PXI-6052E DAQPad-6052E DAQCard-6024E DAQCard-6062E	ND_INT_AI_GND ND_INT_REF_5V ND_INT_CM_REF_5V ND_INT_AO_GND_VS_AI_GND ND_INT_AO_CH_0 ND_INT_AO_CH_0_VS_REF_5V ND_INT_AO_CH_1 ND_INT_AO_CH_1_VS_AO_CH_0 ND_INT_DEV_TEMP ND_INT_AO_CH_1_US_REF_5V
AT-AI-16XE-10 PCI-6032E PCI-6033E NEC-AI-16XE-50	ND_INT_AI_GND ND_INT_REF_5V ND_INT_AO_GND_VS_AI_GND

Table B-2. Valid Internal Analog Input Channels (Continued)

Device	Internal Channels
AT-MIO-16E-1 AT-MIO-16E-2 AT-MIO-64E-3 AT-MIO-16DE-10 AT-MIO-16E-10 DAQPad-6020E NEC-MIO-16E-4	ND_INT_AI_GND ND_INT_REF_5V ND_INT_CM_REF_5V ND_INT_AO_GND_VS_AI_GND ND_INT_AO_CH_0 ND_INT_AO_CH_0_VS_REF_5V ND_INT_AO_CH_1 ND_INT_AO_CH_1_VS_REF_5V
PCI-6023E PCI-6034E DAQCard-6023E	ND_INT_AI_GND ND_INT_REF_5V ND_INT_CH_REF_5V ND_INT_AO_GND_VS_AI_GND ND_INT_AO_CH_0 ND_INT_AO_CH_0_VS_REF_5V ND_INT_DEV_TEMP
DAQCard-AI-16E-4 DAQCard-AI-16XE-50	ND_INT_AI_GND ND_INT_REF_5V

Table B-2. Valid Internal Analog Input Channels (Continued)

Device	Internal Channels
PCI-6110E	ND_INT_AI_GND_AMP_0 ND_INT_AI_GND_AMP_1 ND_INT_AI_GND_AMP_2 ND_INT_AI_GND_AMP_3 ND_INT_AO_CH_0_AMP_0 ND_INT_AO_CH_0_AMP_1 ND_INT_AO_CH_0_AMP_2 ND_INT_AO_CH_0_AMP_3 ND_INT_AO_CH_0_VS_REF_AMP_0 ND_INT_AO_CH_0_VS_REF_AMP_1 ND_INT_AO_CH_0_VS_REF_AMP_2 ND_INT_AO_CH_0_VS_REF_AMP_3 ND_INT_AO_CH_1_AMP_0 ND_INT_AO_CH_1_AMP_1 ND_INT_AO_CH_1_AMP_2 ND_INT_AO_CH_1_AMP_3 ND_INT_AO_CH_1_VS_REF_AMP_0 ND_INT_AO_CH_1_VS_REF_AMP_1 ND_INT_AO_CH_1_VS_REF_AMP_2 ND_INT_AO_CH_1_VS_REF_AMP_3 ND_INT_AO_GND_VS_AI_GND_AMP_0 ND_INT_AO_GND_VS_AI_GND_AMP_1 ND_INT_AO_GND_VS_AI_GND_AMP_2 ND_INT_AO_GND_VS_AI_GND_AMP_3 ND_INT_CM_REF_AMP_0 ND_INT_CM_REF_AMP_1 ND_INT_CM_REF_AMP_2 ND_INT_CM_REF_AMP_3 ND_INT_REF_AMP_0 ND_INT_REF_AMP_1 ND_INT_REF_AMP_2 ND_INT_REF_AMP_3

Table B-2. Valid Internal Analog Input Channels (Continued)

Device	Internal Channels
PCI-6111E	ND_INT_AI_GND_AMP_0 ND_INT_AI_GND_AMP_1 ND_INT_AO_CH_0_AMP_0 ND_INT_AO_CH_0_AMP_1 ND_INT_AO_CH_0_VS_REF_AMP_0 ND_INT_AO_CH_0_VS_REF_AMP_1 ND_INT_AO_CH_1_AMP_0 ND_INT_AO_CH_1_AMP_1 ND_INT_AO_CH_1_VS_REF_AMP_0 ND_INT_AO_CH_1_VS_REF_AMP_1 ND_INT_AO_GND_VS_AI_GND_AMP_0 ND_INT_AO_GND_VS_AI_GND_AMP_1 ND_INT_CM_REF_AMP_0 ND_INT_CM_REF_AMP_1 ND_INT_REF_AMP_0 ND_INT_REF_AMP_1

Table B-3. Internal Channel Purposes for Analog Input Devices

Internal Channel	Purpose
ND_INT_AI_GND	Analog Input Channels Offset
ND_INT_AO_GND_VS_AI_GND	Ground Differential
ND_INT_AO_CH_0	Analog Output Channel 0 Offset/Linearity
ND_INT_AO_CH_1	Analog Output Channel 1 Offset/Linearity
ND_INT_CM_REF_5V	Analog Input Channels Offset
ND_INT_REF_5V	Analog Input Channels Gain
ND_INT_AO_CH_0_VS_REF_5V	Analog Output Channel 0 Gain
ND_INT_AO_CH_1_VS_REF_5V	Analog Output Channel 1 Gain
ND_INT_AO_CH_1_VS_AO_CH_0	Analog Output Channel 1 vs. Analog Output Channel 0
ND_INT_DEV_TEMP	Device Temperature

Internal Channel constants are defined in the following header files:

- C programmers—`NIDAQCNS.H` (`DATAACQ.H` for LabWindows/CVI)
- BASIC programmers—`NIDAQCNS.INC` (Visual Basic for Windows programmers should refer to the *Programming Language Considerations* section in Chapter 1, *Using the NI-DAQ Functions*, for more information.)
- Pascal programmers—`NIDAQCNS.PAS`

Note When the channel is `ND_INT_DEV_TEMP`, you can compute the temperature from the retrieved voltage by applying the following formulas:

- For VXI MIOs:

$$T(°C) = ((Voltage \times 100) - 32) \times 5/9$$

- For all other supported E Series devices:

$$T(°C) = (Voltage \times 100) - 50$$

DAQ Device Gain Settings

Table B-4 lists the valid gain settings for DAQ devices.

Table B-4. Valid Gain Settings

Device	Valid Gain Settings
Most E Series devices	–1 (for a gain of 0.5), 1, 2, 5, 10, 20, 50, 100
All 16XE-50 devices	1, 2, 10, 100
AT-MIO-16X, PCI-MIO-16XE-10, PCI-6031E (MIO-64XE-10), PCI-6032E (AI-16XE-10), PCI-6033E (AI-64XE-10), PXI-6030E, and Lab and 1200 Series devices	1, 2, 5, 10, 20, 50, 100
DAQCard-500/700, 516 and LPM devices	gain is ignored because gain is always 1
PCI-6110E, PCI-6111E	–2 (for gain of 0.2) 1, 2, 5, 10, 20, 50 –1 (for gain of 0.5)
PCI-4451, PCI-4452, NI 4551, NI 4552	–20, –10, 0, 10, 20, 30, 40, 50, 60 (dB)
PCI-4453, PCI 4454	0 (dB)
6023E, 6024E, 6025E, 6034E, 6035E,	–1 (for a gain of 0.5) 1, 10, 100

Voltage Calculation

`AI_VScale` and `DAQ_VScale` calculate voltage from **reading** as follows:

$$\text{voltage} = \left(\frac{\textbf{reading} - \textbf{offset}}{\textbf{maxReading}}\right) \times \left(\frac{\textbf{maxVolt}}{\textbf{gain} \times \textbf{gainAdjust}}\right)$$

where:

- **maxReading** is the maximum binary reading for the given board, channel, range, and polarity.
- **maxVolt** is the maximum voltage the board can measure at a gain of 1 in the given range and polarity.

Table B-5 lists the values of **maxReading** and **maxVolt** for different boards.

Table B-5. The Values of maxReading and maxVolt

Device	Unipolar Mode		Bipolar Mode	
	maxReading	**maxVolt**	**maxReading**	**maxVolt**
Most E Series devices	4,096	10 V	2,048	5 V
16-bit E Series devices	65,536	10 V	32,768	10 V
Lab-PC+, Lab-PC-1200, Lab-PC-1200AI, DAQPad-1200, DAQCard-1200, PCI-1200	4,096	10 V	2,048	5 V
DAQCard-700, LPM devices	4,096	*	2,048	*
516 devices	—	—	32,768	5 V
DAQCard-500	—	—	2,048	5 V
PCI-6110E and PCI-6111E	—	—	2,048	10 V
DSA devices	—	—	2,147,418,112	10 V

* The value of **maxVolt** depends on **inputRange**, as discussed in `AI_Configure`.

For the PC-LPM-16 and DAQCard-1200, gain is ignored, and the following formula is used:

$$\text{voltage} = \left(\frac{\textbf{reading} - \textbf{offset}}{\textbf{maxReading}}\right) \times (\textbf{maxVolt})$$

Offset and Gain Adjustment

Measurement of Offset

To determine the **offset** parameter used in the `AI_VScale` and `DAQ_VScale` functions, follow this procedure:

1. Ground analog input channel *i*, where *i* can be any valid input channel.
2. Call the `AI_Read` function with **gain** set to the gain that will be used in your real acquisition (*g*). The reading given by the `AI_Read` function is the value of **offset**. The offset is only valid for the gain setting at which it was measured. Remember that the data type of **offset** in the `AI_VScale` and `DAQ_VScale` functions is floating point, so if you use `AI_Read` to get the offset, you have to typecast it before passing it to the scale function.

Note Another way to read the offset is to perform multiple readings using a DAQ function call and average them to be more accurate and reduce the effects of noise.

Measurement of Gain Adjustment

To determine the **gainAdjust** parameter used in the `AI_VScale` and `DAQ_VScale` functions, follow this procedure:

1. Connect the known voltage V_{in} to channel *i*.
2. Call the `AI_Read` function with gain equal to *g*. Use the reading returned by `AI_Read` with the offset value determined above to calculate the real gain.

Note You can use the DAQ functions to take many readings and average them instead of using the `AI_Read` function.

The real gain is computed as follows:

$$G_R = \left(\frac{\textbf{reading} - \textbf{offset}}{\textbf{maxReading}}\right) \times \left(\frac{\textbf{maxVolt}}{V_{in}}\right)$$

The gain adjustment is computed as follows:

$$\textbf{gainAdjust} = \left[1 - \frac{(g - G_R)}{g}\right]$$