United Electronic Industries

The High-Performance Alternative

UeiDaq Framework LabVIEW User Manual

February 2015 Edition

1. Introduction

LabVIEW is a development environment based on graphical programming. The UeiDaq VI Library, a series of virtual instruments (VIs) for using LabVIEW with your PowerDNA or PowerDAQ device, is included with the PowerDNA and PowerDAQ software.

Before you start building your application, you must be familiar with the following UeiDaq framework concepts:

- Location of the UeiDaq examples.
- Location of the UeiDaq VIs in LabVIEW.
- UeiDaq Refnum.
- Session configuration.
- Timing configuration.
- Trigger configuration.
- Session control.
- Data transfer.
- Property node.
- Error handling.

2. Location of the UeiDaq examples

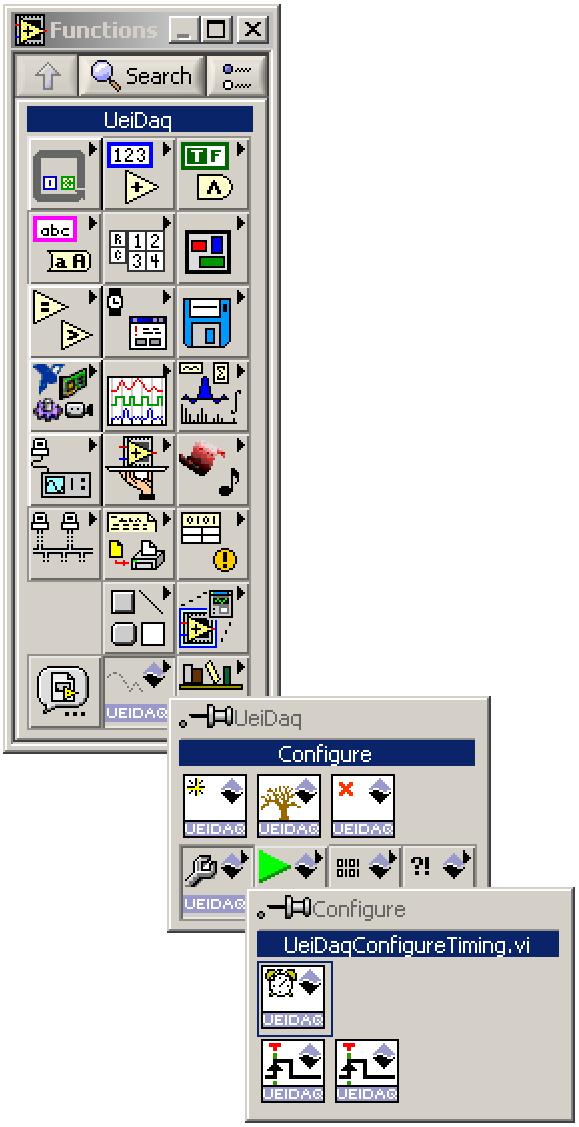
The UeiDaq examples illustrate some common applications. You can find these examples in the LabVIEW\examples\UeiDaq directory for LabVIEW (usually installed at c:\program files\national instruments\LabVIEW xxxx\examples\UeiDaq).

3. Location of the UeiDaq VIs in LabVIEW

You can find the UeiDaq VIs in the Functions palette from your LabVIEW block diagram or in the LabVIEW\vi.lib\UeiDaq directory.

If the UeiDaq Function palette, is not visible, click on the menu item “Tools/Options…”, select the item “Controls/Functions Palettes” and select the item “UeiDaq” in the “Palette View” control.

You can also add the UeiDaq function palette to your existing custom palette. Click on the menu item “Tools/Advanced/Edit Palette Views…”. Right-click on the palette where you want to insert the UeiDaq palette and select “Insert Submenu…”, click on the option “Link to an existing menu file” and browse to the file LabVIEW\menus\ueidaq\UEI.

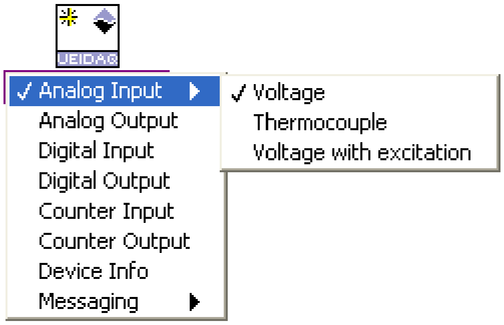


4. UeiDaq Refnum

“UeiDaq Refnum In” is a unique identifier that specifies the device and sub-system used for the acquisition. This identifier is produced by the UeiDaqCreate**Session VI and used as an input to all other UeiDaq VIs and the property node. The UeiDaq VIs use “UeiDaq Refnum Out”, which is identical to “UeiDaq Refnum In”, to simplify dataflow programming. “UeiDaq Refnum Out” is similar to the duplicate file sessions provided by the file I/O VIs.

5. *Session Configuration*

The Session manages communication with a data acquisition device subsystem. In order to do anything with your data acquisition device, you need to create a session specifying which subsystem of the device you wish to use (Analog Input, Analog Output, Digital Input…).
The UeiDaq VI library uses polymorphic VIs to let you select the session type.



All instances of "UeiDaqCreateSession.vi" have one parameter in common: the resource string.

The UeiDaq VI library 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>

Available device classes are:

- "pwrdaq" for PowerDAQ PCI and PXI boards
- "pdna" for PowerDNA Ethernet I/O modules
- "simu" implements software simulation of various data acquisition devices, allowing you to start programming the framework without hardware.

The device ID must start with the string "dev" followed by the 0 based device number.

The IP address needs to be specified only if the device class requires it. PowerDNA devices need an IP address; PowerDAQ devices will ignore it if specified.

The subsystem must be one of the following:

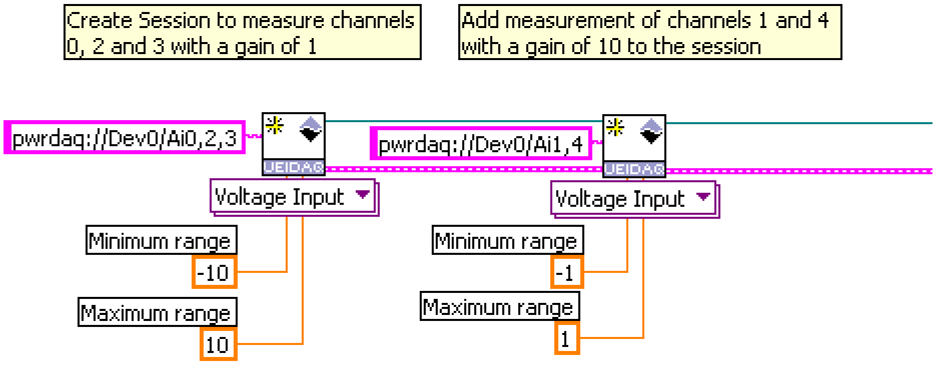
- AI: analog input session to measure voltage, temperature or strain.
- AO: analog output session to generate voltage.
- DI: digital input session to measure discrete signals.
- DO: digital output session to generate digital patterns.
- CI: counter input session to count discrete events, or measure pulse width and period.
- CO: counter output session to generate pulses and pulse trains.
- CAN: CAN bus session to send/receive data over a CAN bus.
- COM: serial port session to send/receive data over a serial port.
- ATX: ARINC-429 transmitter session to send data to an ARINC-429 bus.
- ARX: ARINC-429 receiver session to receive data from an ARINC-429 bus.
- MIL: MIL-1553 bus session to send/receive data over a MIL-1553 bus as a bus monitor, Remote Terminal or Bus Controller.
- IRIG: Irig session to receive and/or generate timing signal using the IRIG time code format.
- HDLC: session to send/receive data over a synchronous serial link using HDLC protocol.

The channel list is a comma-separated list of channels; you can also specify a range of channels using the lowest and highest channels separated by a colon. The channels in the channel list do not have to be sequential; you can even repeat channels multiple times.

The resource string is not case sensitive; here are a few examples:

```
Pwrdaq://Dev0/Ai0:15
PDNA://192.168.100.2/Dev1/AO3
simu://Dev2/Di3
```

All channels included in the resource string will share the same parameters such as gain or input mode. To configure multiple channels with different parameters within one session you need to call :"UeiDaqCreateSession.vi" multiple times feeding the refnum obtained from the first call into the subsequent calls.

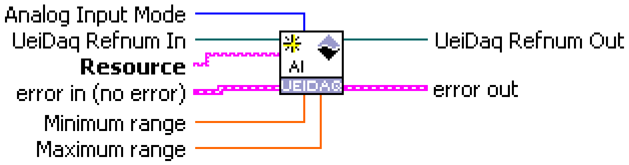


5.1. Analog Input

5.1.1. Voltage

This session performs voltage measurements.

Use the **AnalogInput:Voltage** instance of **UeiDaqCreateSession.vi**



Minimum range The minimum voltage you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Maximum range The maximum voltage you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Analog Input Mode The input mode of the Analog input(s). Possible values are 'Single-Ended' or 'Differential'.

Built-In Test (BIT)

Some AI devices are capable of connecting and internal voltage reference to each input channel.

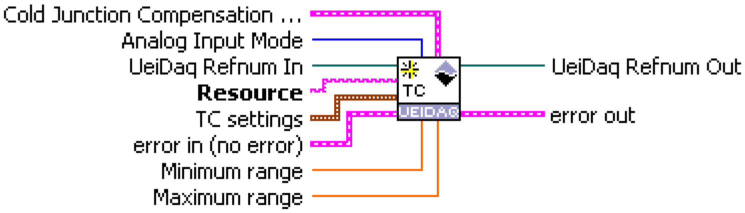
TO BE DONE

5.1.2. Current

5.1.3. Thermocouple

This session performs temperature measurements. It measures thermocouple voltages and linearizes those voltages to temperatures using ITS-90 coefficient table.

Use the **AnalogInput:Thermocouple** instance of **UeiDaqCreateSession.vi**



Minimum range The minimum temperature you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Maximum range The maximum temperature you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Analog Input Mode The input mode of the Analog input(s). Possible values are 'Single-Ended' or 'Differential'.

Cold Junction Compensation Settings Specifies the cold-junction compensation settings:

- **Cold Junction compensation type**: The type of CJC :
  'builtin' uses the temperature sensor on the STP-AIU to measure the CJ temperature
  'constant' use a constant value as CJ temperature
  'channel' uses one of the input channels to measure temperature

- **Cold Junction Compensation temperature**: the CJ temperature constant
- **Cold Junction Compensation channel:** the channel used to measure temperature

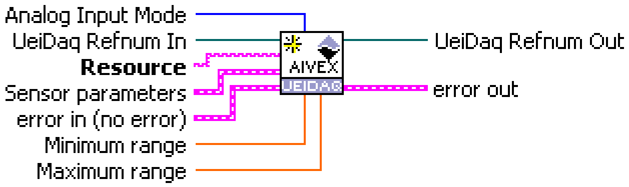
TC settings Specifies the thermocouple type and the temperature scale:

- **Thermocouple type:** Select your TC type ‘E’, ‘K’, ‘J’ etc…
- **Temperature Scale:** Select the scale used to convert TC voltage to temperature: ‘celsius’,’kelvin’ or ‘farenheit’

5.1.4. Voltage with excitation

This session performs voltage measurements on wheatstone bridge circuits. The bridge might be built-in in a load cell or assembled with strain gauges.
The AI device must be capable of providing excitation to power the bridge.

Use the **AnalogInput:Voltage With Excitation** instance of **UeiDaqCreateSession.vi**



Minimum range The minimum voltage you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Maximum range The maximum voltage you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Analog Input Mode The input mode of the Analog input(s). Possible values are 'Single-Ended' or 'Differential'.

Sensor parameters The parameters for the sensors attached to the channels.

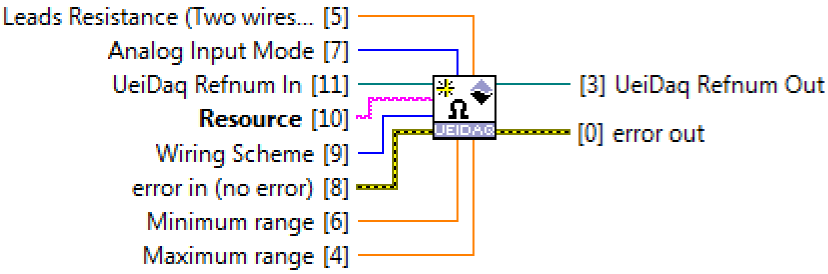
- **Sensor bridge type:** The type of bridge ‘quarter’, ‘half’ or ‘full’
- **Excitation voltage:** The excitation voltage used to power the bridge
- **Scale data with excitation:** Specifies whether you wish to use ratiometric measurements (measured voltage is divided by excitation to return mV/V)

5.1.5. Resistance

This session performs resistance measurements. The AI device must be capable of measuring resistance directly or must be associated with a terminal block that includes resistance measurement circuitry.

Measurements are returned in Ohms.

Use the **AnalogInput:Resistance** instance of **UeiDaqCreateSession.vi**



Minimum range The minimum resistance you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Maximum range The maximum resistance you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Analog Input Mode The input mode of the Analog input(s). Possible values are 'Single-Ended' or 'Differential'.

Wiring Scheme: The wiring scheme used to connect the resistive sensor to the channel. Possible values are ‘2 wires’, ‘3 wires’, ‘4 wires’.

Leads Resistance: The resistance of the leads connecting the resistive sensor to the AI device. Only needed in two wires mode.

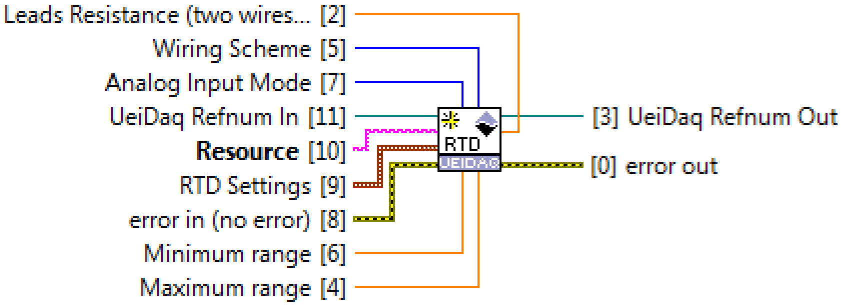
5.1.6. RTD

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

The AI device must be capable of measuring resistance directly or must be associated with a terminal block that includes resistance measurement circuitry.

Measurements are returned as temperatures.

Use the **AnalogInput:RTD** instance of **UeiDaqCreateSession.vi**



Minimum range The minimum temperature you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Maximum range The maximum temperature you expect to measure.
If your device has programmable gains, the gain will be automatically adjusted to match the specified input range as close as possible.

Analog Input Mode The input mode of the Analog input(s). Possible values are 'Single-Ended' or 'Differential'.

Wiring Scheme: The wiring scheme used to connect the resistive sensor to the channel. Possible values are ‘2 wires’, ‘3 wires’, ‘4 wires’.

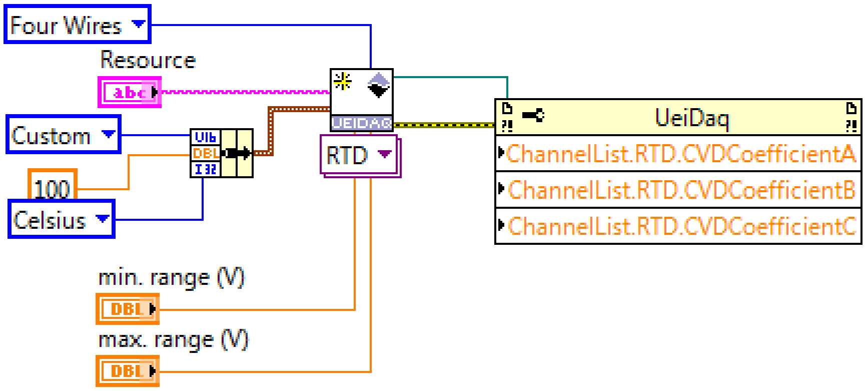
Leads Resistance: The resistance of the leads connecting the RTD sensor to the AI device. Only needed in two wires mode.

RTD Settings: Specifies the type of RTD connected to the AI channel

- **RTD type:** RTD sensors are specified using the "alpha" (α) constant. It is also known as the temperature coefficient of resistance, which defines the resistance change factor per degree of temperature change. The RTD type is used to select the proper coefficients A, B and C for the Callendar Van-Dusen equation, which is used to convert resistance measurements to temperature.
- **RTD nominal resistance at 0 degC:** For example 100 for PT100 RTDs.

- **Temperature Scale:** Select the scale used to convert RTD resistance to temperature: ‘celsius’,’kelvin’ or ‘farenheit’

Custom RTD types are supported. You can specify the Callendar Van-Dusen coefficients after having called this VI using the UeiDaq Property node:

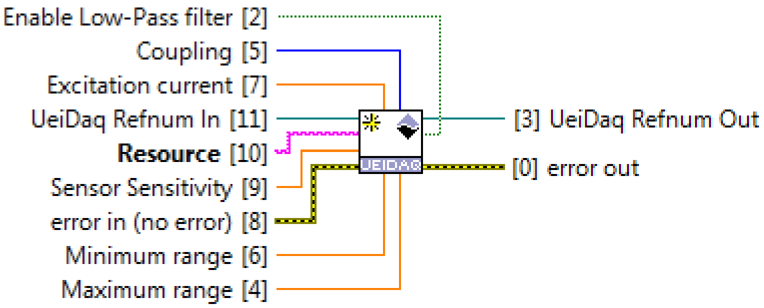

Four Wires
Resource
abc
Custom
100
Celsius
UEIDAQ
RTD
UeiDaq
ChannelList.RTD.CVDCoefficientA
ChannelList.RTD.CVDCoefficientB
ChannelList.RTD.CVDCoefficientC
min. range (V)
DBL
max. range (V)
DBL

5.1.7. Accelerometer

This session performs acceleration and vibration measurements from ICP or IEPE sensors.

The AI device must be capable of providing current excitation for the sensor.

Use the **AnalogInput:Accelerometer** instance of **UeiDaqCreateSession.vi**



Low Limit: the minimum expected measurement. The unit is the same as the sensor sensitivity unit. The low limit combined with the high limit determines which gain will be configured.

High Limit: the maximum expected measurement. The unit is the same as the sensor sensitivity unit. The low limit combined with the high limit determines which gain will be configured

Sensor Sensitivity: The sensor sensitivity. Its unit determines the unit of the measurements. For example, if the sensitivity is specified in mV/g, the measurements will be returned in number of g.

Excitation Current: the excitation current used to power the IEPE sensor.

Coupling: Configures the high-pass filter. DC coupling disables the filter, AC coupling enables a 0.1Hz high-pass filter.

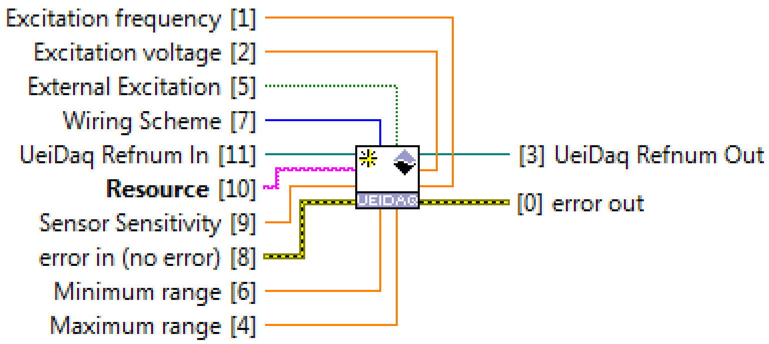
Low pass Filter: Enables or disables the low-pass analog filter. Cut off frequency is 48kHz.

5.1.8. LVDT/RVDT

This session performs displacement measurements from LVDT or RVDT sensors.

The AI device must be capable of providing excitation for the sensor.

Use the **AnalogInput:LVDT/RVDT** instance of **UeiDaqCreateSession.vi**



Minimum range: minimum value you expect to measure. The unit is the distance unit specified by your LVDT sensitivity. For example, if your LVDT sensitivity is given in mV/V/mm, the distance unit is mm. Similarly for RVDTs the unit is the angle unit used in the sensitivity.

Maximum range: maximum value you expect to measure. The unit is the position unit specified by your LVDT sensitivity. For example if your LVDT sensitivity is given in mV/V/mm, the distance unit is mm. Similarly for RVDTs, the unit is the angle unit used in the sensitivity.

Sensor Sensitivity: the sensor sensitivity specified in mV/V/<position unit>. The position unit specifies the unit of the min. and max. range as well as the unit of the data read from the input channels.

Wiring Scheme: the wiring scheme used to connect the LVDT/RVDT to the channel. Possible values are ‘4 wires’ or ‘5 wires’

Excitation voltage: the amplitude in volt RMS of the excitation sine wave.

Excitation frequency: the frequency of the excitation sine wave in Hertz.

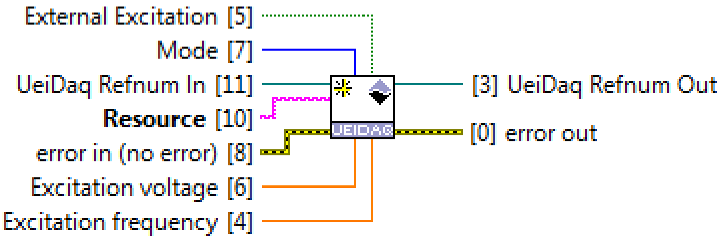
External excitation: specifies whether you wish to provide external excitation or use the excitation provided by the AI device.

5.1.9. Synchro/Resolver

This session performs displacement measurements from Synchro or Resolver sensors.

The AI device must be capable of providing excitation for the sensor.

Use the **AnalogInput:Synchro/Resolver** instance of **UeiDaqCreateSession.vi**



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 in Hertz.

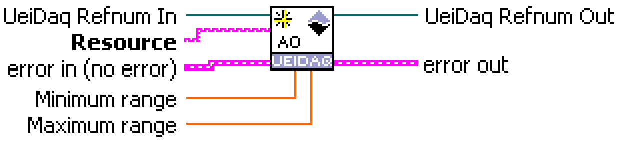
External Excitation: specifies whether you wish to provide external excitation or use the excitation provided by the AI device.

5.2. Analog Output

5.2.1. Voltage

This session performs voltage output.

Use the **Analog Output./Voltage** instance of **UeiDaqCreateSession.vi**



Minimum range: minimum value of the voltage to generate. This value is ignored on AO devices with a fixed output range.

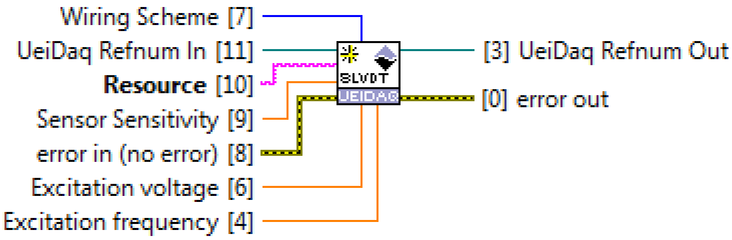
Maximum range: maximum value of the voltage to generate. This value is ignored on AO devices with a fixed output range.

Most AO devices come with a fixed output range (for example -10V/+10V)

5.2.2. Simulated LVDT/RVDT

This session simulates LVDT or RVDT sensor output.

Use the **Analog Output./Simulated LVDT/RVDT** instance of **UeiDaqCreateSession.vi**



Sensor Sensitivity: the sensor sensitivity specified in mV/V/<position unit>. The position unit specifies the unit of the data written to the output channels.

Wiring Scheme: the wiring scheme used to connect the simulated LVDT/RVDT to the measurement device. Possible values are ‘4 wires’ or ‘5 wires’

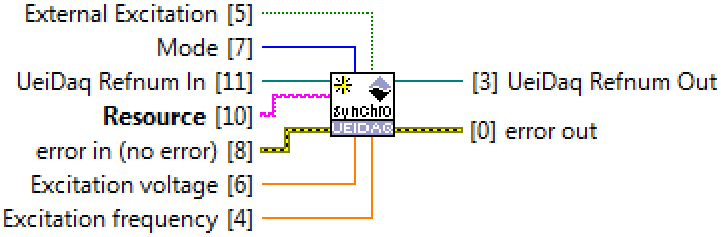
Excitation voltage: the amplitude in volt RMS of the excitation sine wave.

Excitation frequency: the frequency of the excitation sine wave in Hertz.

5.2.3. Simulated Synchro/Resolver

This session simulates LVDT or RVDT sensor output.

Use the **Analog Output./Simulated Synchro/Resolver** instance of **UeiDaqCreateSession.vi**



Sensor Mode: the type of sensor (synchro or resolver) to simulate.

Excitation Voltage: the amplitude of the excitation sine wave in volts RMS.

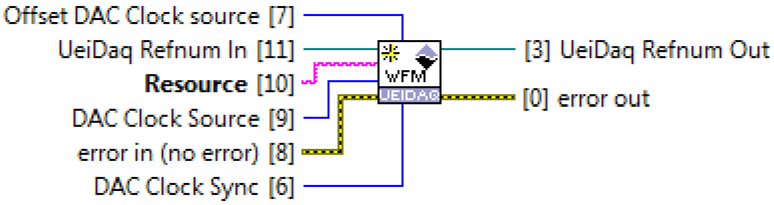
Excitation Frequency: the frequency of the excitation sine wave in Hertz.

External Excitation: specifies whether to use internal excitation or use external excitation provided by the measurement device.

5.2.4. Waveform

This session generates a waveform out of an arbitrary waveform output device.

Use the **Analog Output./Waveform** instance of **UeiDaqCreateSession.vi**



DAC clock source: The source of the clock used to time the main DAC:

- `SYNC2`: use SYNC2 line for clock
- `SYNC0`: use SYNC0 line for clock
- `ALT0`: use layer channel zero PLL routed to Channel 0 trigger out to clock all channels
- `TMR`: clock from internal TMR0 timebase
- `DIO0`: use channel DIO0 line for clock
- `PLL`: clock from PLL
- `SW`: DAC is clocked by software (DC offset only)

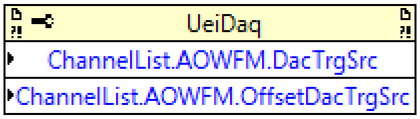
Offset DAC clock source: The source of the clock used to time the offset DAC:

- `DIO0`: use channel DIO0 line for clock
- `DIO1`: use channel DIO1 line for clock
- `DAC`: main DAC clock divided is the source of offset DAC
- `PLL`: PLL is the source of the offset DAC (independent of main DAC)
- `SW`: offset DAC is clocked by software (DC offset only)

Main DAC clock sync: Specifies where a clock signal should be routed to synchronize with other channels and/or layers.
Route your signal out to TrgOut to synchronize multiple channels on the same AO-364.
Route your signal out to the Sync lines to synchronize multiple AO-364s.
Only valid for channel 0 on AO-364:

- `None`: No sync routing
- `DIO1ToTrgOut`: Route the DIO1/trigger input pin to Channel0 trigger out (channel 0 only)
- `DIO0ToTrgOut`: Route the DIO0/clock input pin to Channel0 trigger out (channel 0 only)
- `PLLToTrgOut`: Route PLL clock to Channel0 trigger out (channel 0 only)`UeiAOWaveformClockRoutePLLToSYNC2`: Route PLL clock to SYNC2 (channel 0 only)
- `PLLToSYNC0`: Route PLL clock to SYNC0 (channel 0 only)

In addition you can set additional parameters using the UeiDaq property node:



UeiDaq
ChannelList.AOWFM.DacTrgSrc
ChannelList.AOWFM.OffsetDacTrgSrc

Main DAC trigger source: source used to trigger a new period out of the main DAC:

- `None`: no trigger, layer outputs when clock is available

- `CH0`: channel 0 will deliver clock triggered on CH0_TRIGIN line
- `SYNC3`: use SYNC3 line as a trigger
- `SYNC1`: use SYNC1 line as a trigger

Offset DAC trigger source: source used to trigger a new period out of the offset DAC:

- `None`: no trigger, layer outputs when clock is available (use with NIS clocking)
- `SYNC3`: use SYNC3 line as a trigger
- `SYNC1`: use SYNC1 line as a trigger
- `ALT0`: use channel 0 CH0-TIN line for trigger (needs to be connected to a source)
- `DIO1`: use channel DIO1 line for trigger
- `SW`: use software trigger (simultaneous only within single layer)

5.2.5. Protected

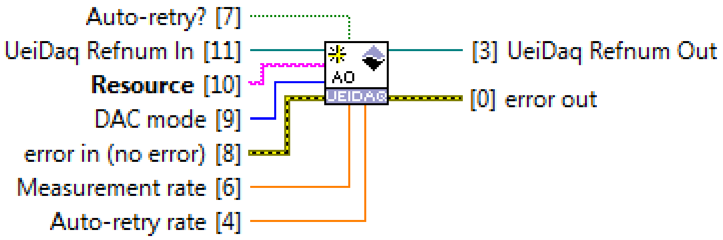
This session generates voltage out of a Guardian analog output device.

Guardian AO devices are capable of measuring up to 5 diagnostic measurements and monitoring up to 3 of those measurement measurements per AO channel:

- Current flowing through the AO channel
- Output Voltage
- Temperature of the DAC

Each AO channel comes with two DACs and two circuit breakers associated with each DAC.
Each circuit breaker can monitor at most three of the five diagnostic channels and open the circuit when one of the diagnostic measurements is off limit. The circuit breaker can periodically attempt to re-connect the circuit

Use the **Analog Output./Protected** instance of **UeiDaqCreateSession.vi**



Auto-retry?: specifies whether the AO device attempts to close the circuit after an over or under limit condition was detected.

DAC mode: Selects which DAC if any is connected to the output channels. Possible values are 'Disconnected', 'Both DACs connected', 'DAC A connected', 'DAC B connected'

Measurement rate: the rate at which the AO device monitors diagnostic channels. This rate has a direct influence on how fast the device reacts to an under or over-current condition.

Auto-retry rate: specifies how often the AO device attempts to close the circuit.

Diagnostic Measurements

Diagnostic measurements are read using an Analog Input session. Up to five analog input channels can be associated with each analog output channel.
AI channels associated with AO channel 0 are 0,1,2,3,4
AI channels associated with AO channel 1 are 8,9,10,11,12
Etc…

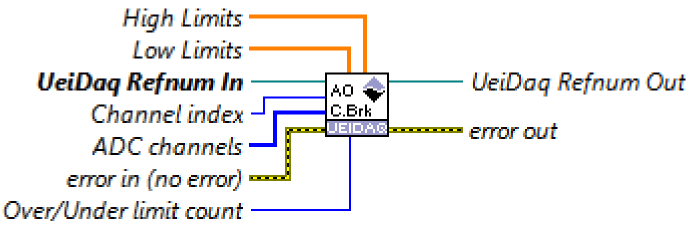
You can read all 40 diagnostic measurements with channel list:
"0:4,8:12,16:20,24:28,32:36,40:44,48:52,56:60"

Circuit Breaker

The three diagnostic measurements are set by default to:

- Current limited to +/-15mA
- Voltage limited +/-12V
- Temperature limited to -50c/+105c.

You can modify those parameters with the VI **UeiDaqSetAOcircuitbreaker.vi**



High limits: Array containing each diagnostic high limit. The circuit will open when any of the diagnostic goes above this limit.

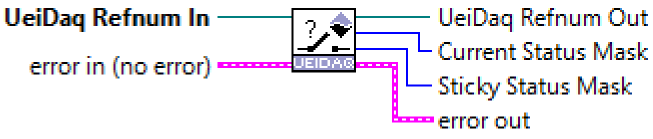
Low limits: Array containing each diagnostic low limit. The circuit will open when any of the diagnostic goes below this limit.

Channel Index: Specify which AO channel will be re-configured.

ADC channels: An array containing up to 5 values to select the diagnostic measurements of each channel. 0:current, 1:DAC A Voltage, 2: DAC B voltage, 3: output voltage, 4: temperature, 5:none

Over/Under limit count: Specifies the number of consecutive over/under limit diagnostic readings that must occur in order to trip the circuit breaker.

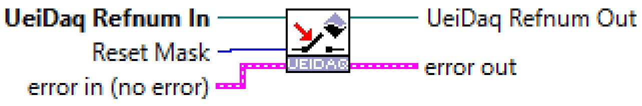
You can monitor the circuit breaker status with the VI **UeiDaqGetCircuitBreakerStatus.vi**.



Current Status Mask: Each bit of this status mask represents the state of a circuit breaker. A bit set to 1 means that the breaker is currently tripped.

Sticky Status Mask: Each bit of this status mask represents the sticky state of a circuit breaker. A bit set to 1 means that the breaker was tripped at least once since the last time this VI was called.

When auto-retry is disabled, you can manually reset a tripped circuit breaker with the VI **UeiDaqResetCircuitBreaker.vi.**



Reset Mask: Each bit in the reset mask specifies which circuit breaker(s) to reset (1 to reset, 0 to leave alone)

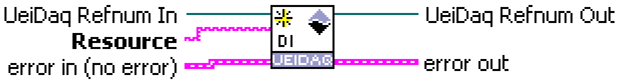
5.3. Digital Input

5.3.1. Input

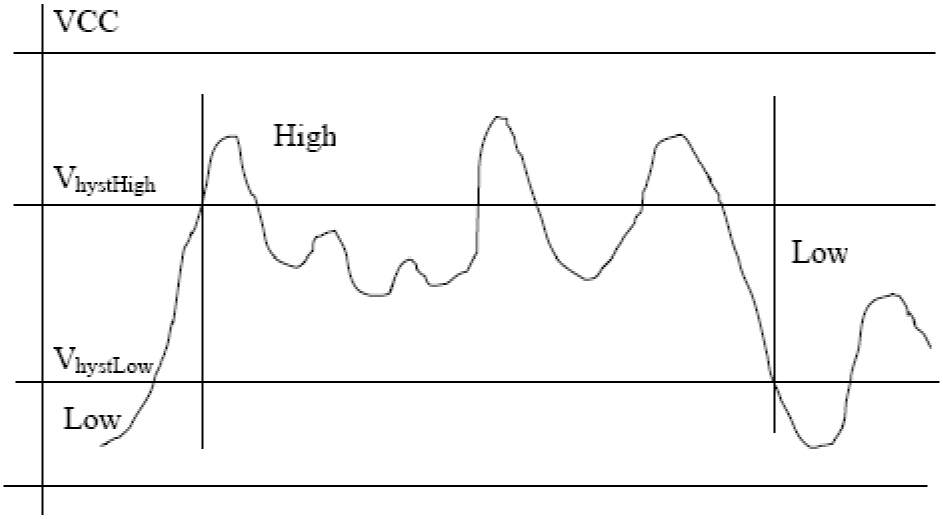
This session performs digital input.

Digital input lines are organized in ports. It is only possible to read an entire port.

Use the **Digital Input:Input** instance of **UeiDaqCreateSession.vi**



Some digital input devices support programmable hysteresis threshold:
An input line will switch from low to high state only if the input voltage level goes above the high threshold.
An input line will switch from high to state state only if the input voltage level goes below the low threshold.



Use the UeiDaq property node to set the threshold limits.

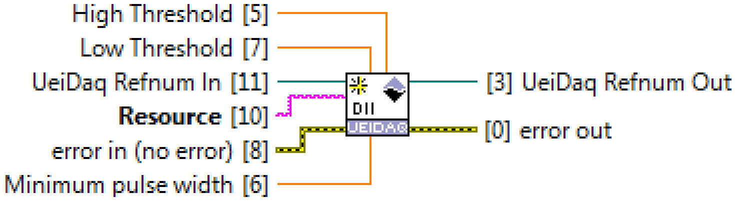
5.3.2. Industrial

This session reads digital inputs of a Guardian digital input device.

Digital input lines are organized in ports. It is only possible to read an entire port

Guardian DI devices are capable of measuring voltages flowing through each of the input lines.

Use the **Digital Input:Industrial** instance of **UeiDaqCreateSession.vi**



Low Threshold: the low hysteresis threshold.

High Threshold: the high hysteresis threshold.

Minimum pulse width: the minimum pulse width in ms. Any pulse that is smaller than this value will not change the input state. Use 0.0 to disable the digital input filter.

This VI configures all input lines in the port specified in the resource string with the same parameters. You can configure each line individually by specifying them in the resource string:

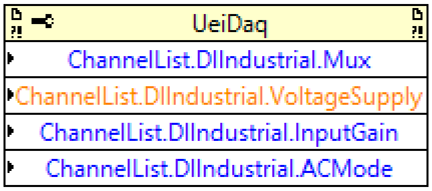
- **pdna://192.168.100.2/dev0/di0** configures all lines in port 0
- **pdna://192.168.100.2/dev0/diline0:7** only configures lines 0 to 7

Diagnostic Measurements

Voltage measurements can be read using an Analog Input session. One analog input channel is associated with each digital input line. Measurements unit is Volt (V).

Built-In Test (BIT)

Some industrial devices (DIO-449 for example) support advanced features through channel the UeiDaq property node.



UeiDaq
ChannelList.DIIndustrial.Mux
ChannelList.DIIndustrial.VoltageSupply
ChannelList.DIIndustrial.InputGain
ChannelList.DIIndustrial.ACMode

Signal injection mux: state of the signal injection multiplexer

- **Tristate**: set the mux to tri-state mode to disconnect voltage supply from input line
- **Diagnostic**: set mux to Diagnostic mode to connect internal voltage source to each input line in order to test that it is functional
- **Pull-up**: setting mux to pull-up mode connects a pull-up resistor between the internal voltage source and the input line to monitor switches or contacts without external circuitry.

Voltage Supply: voltage applied by the internal source.

Input Gain: the input gain used by the ADC to measure the input voltage DC or RMS level.

AC/DC mode: in AC mode the RMS voltage measured at each input is compared with low & high thresholds to determine the state of the input

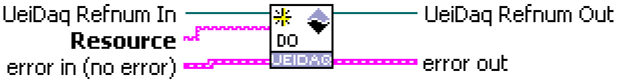
5.4. Digital Output

5.4.1. Output

This session performs digital output.

Digital output lines are organized in ports. It is only possible to write an entire port.

Use the **Digital Output:Output** instance of **UeiDaqCreateSession.vi**



5.4.2. Protected

This session performs digital output on a Guardian digital output device: DIO-432, DIO-433, and relay devices DIO-462 and DIO-463.

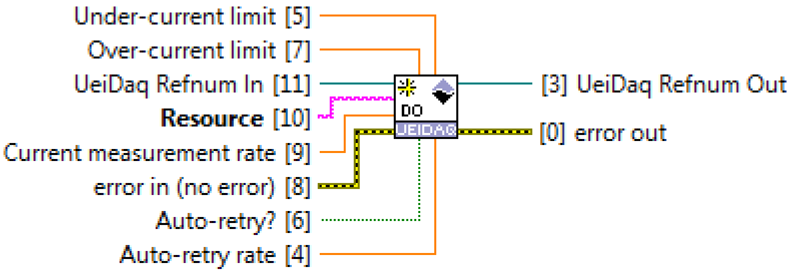
Digital output lines are organized in ports. It is only possible to write an entire port.

Guardian DO devices are capable of measuring and monitoring current flowing through any of the output lines.
In addition Relay devices are capable of measuring Normally Open (when relay is off) and Normally Close (when relay is on) voltages as well as relay temperatures.

Each DO line comes with a circuit breaker that can open the circuit when the current measurement is off limit. The circuit breaker can be manually reset or it can periodically attempt to re-connect the circuit

On some devices (DIO-432 and DIO-433), each DO line is also capable of generating a programmable PWM when switching from low to high or high to low (soft-start or soft-stop)

Use the **Digital Output:Protected** instance of **UeiDaqCreateSession.vi**



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 DO device monitors current. This rate has a direct influence on how fast the line reacts to an under or over-current condition.

Auto-retry?: specifies whether the output line attempts to close the circuit after an over or under current condition.

Auto-retry rate: specifies how often the output line attempts to close the circuit.

This VI configures all output lines in the port specified in the resource string with the same parameters. You can configure each line individually by specifying them in the resource string:

- **pdna://192.168.100.2/dev0/do0** configures all lines in port 0
- **pdna://192.168.100.2/dev0/doline0:7** only configures lines 0 to 7

Diagnostic Measurements

Diagnostic measurements can be read using an Analog Input session.

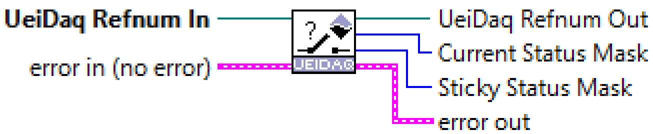
On digital output devices (DIO-432 and 433), one analog input channel is associated with each digital output line. Measurements unit is Ampere (A).

On relay devices (DIO-462 and 463), five analog input channels are associated with each relay.
AI sub-channels associated with relay channel 0 are 0,1,2,3,4
AI sub-channels associated with relay channel 1 are 8,9,10,11,12
Etc…

First AI sub-channel reads Normally Open voltage
Second AI sub-channel reads AC current on common pin (in Amps)
Third AI sub-channel reads DC current on common pin (in Amps)
Fourth AI sub-channel reads Normally Close voltage
Fifth AI sub-channel reads relay temperature (deg. C)

Circuit Breaker

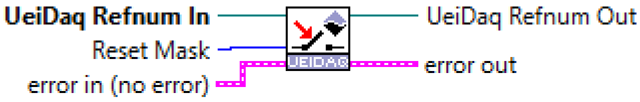
You can monitor the circuit breaker status with the VI **UeiDaqGetCircuitBreakerStatus.vi**.



Current Status Mask: Each bit of this status mask represents the state of a circuit breaker. A bit set to 1 means that the breaker is currently tripped.

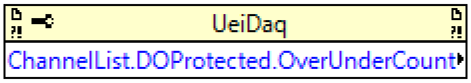
Sticky Status Mask: Each bit of this status mask represents the sticky state of a circuit breaker. A bit set to 1 means that the breaker was tripped at least once since the last time this VI was called.

When auto-retry is disabled, you can manually reset a tripped circuit breaker with the VI **UeiDaqResetCircuitBreaker.vi.**



Reset Mask: Each bit in the reset mask specifies which circuit breaker(s) to reset (1 to reset, 0 to leave alone)

You can set the number of time current measurements are allowed to be off limit before tripping a circuit breaker (default is 1).
Use the UeiDaq property node.

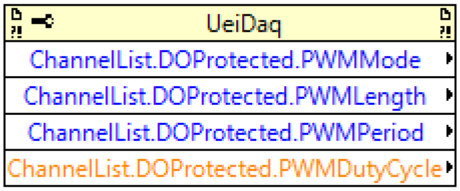


UeiDaq
ChannelList.DOProtected.OverUnderCount

Over/Under Count: Specifies how many consecutive times the monitored current must be out of limits before tripping the circuit breaker.

PWM

The pulse width modulation features are programmable on a per-output line basis using the UeiDaq property node.



UeiDaq
ChannelList.DOProtected.PWMMode
ChannelList.DOProtected.PWMLength
ChannelList.DOProtected.PWMPeriod
ChannelList.DOProtected.PWMDutyCycle

PWM mode: The PWM mode. Possible values are:

- 0: Disable pulsed transitions
- 1: Use a pulse train for low to high transitions (soft-start)
- 2; Use a pulse train for high to low transition (soft-stop)
- 3: Use a pulse train for both transitions (soft-start and soft-stop)
- 4: Continuously output a pulse train
- 5: Output pulse train only when output line is high

PWM length: The length of soft-start and soft-stop PWMs in micro-seconds

PWM period: The period of soft-start and soft-stop PWMs in micro-seconds

PWM duty-cycle: The duty-cycle of soft-start and soft-stop PWMs (value between 0 and 1)

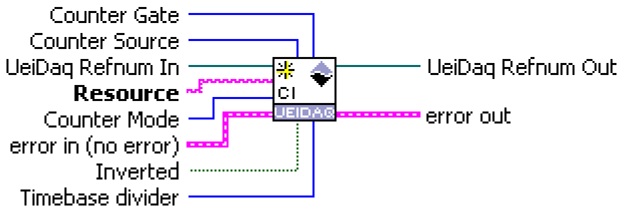
5.5. Counter Input

5.5.1. Input

This session performs count, period and pulse-width measurements.

Measurements are returned as multiples of the timebase.
For example when using the internal 66MHz timebase of a CT-601, a period measurement of 10000 correspond to 10000/66E+6 = 151.15 usecs

Use the **Counter Input:Input** instance of **UeiDaqCreateSession.vi**



Counter Source The source of the signal counted or used as a timebase

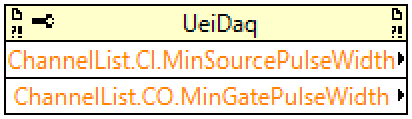
Counter Gate The signal that specify whether the counter/timer is on or off

Counter Mode The mode that specify whether the session counts events, measures a pulse width or measures a period on the signal configured as the source

Inverted Specifies whether the signal at the source is inverted before performing the counting operation

Timebase divider The divider used to scale the timebase speed

Some counter input devices come with digital input filters to reject noisy pulses that are below a programmable width. You can set those filters using the UeiDaq property node:



UeiDaq
ChannelList.CI.MinSourcePulseWidth
ChannelList.CO.MinGatePulseWidth

Min. Source Pulse width: The minimum pulse width in milliseconds detectable at the source input

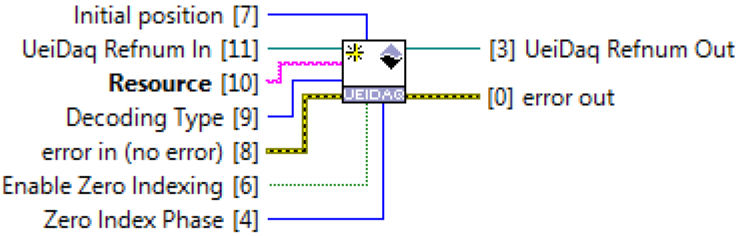
Min. Gate Pulse width: The minimum pulse width in milliseconds detectable at the gate input

5.5.2. Quadrature Encoder

This session performs Quadrature encoder position measurements.

It is designed to work with a device capable of decoding the A, B and optionally Z signals coming out of a quadrature encoder.

Use the **Counter Input:Quadrature Encoder** instance of **UeiDaqCreateSession.vi**



Initial position: The initial number of pulses when measurement begins. This value is reloaded each time the Z input resets the measurement.

Decoding type: The method used to count and interpret the pulses the encoder generates on input A and B. 2x and 4x decoding are more sensitive to smaller changes in position. Possible values are '1x', '2x' and '4x'.

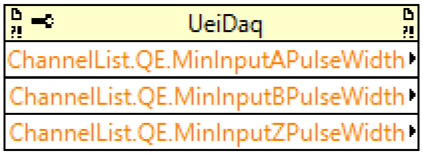
Enable zero indexing: Enable (1) or disable (0) zero indexing. When enabled, Zero indexing resets the measurement to the initial position when the Z, A and B inputs are in a state specified by 'zero index phase'.

Zero index phase: The states at which A and B input signals must be in when Z is high to trigger a measurement reset. Possible values are:

- Z high
- Z high/A low/B low

- Z high/A low/B high
- Z high/A high/B low
- Z high/A high/B high

Quadrature encoder input devices come with digital input filters to reject noisy pulses that are below a programmable width. You can set those filters using the UeiDaq property node:


UeiDaq
ChannelList.QE.MinInputAPulseWidth
ChannelList.QE.MinInputBPulseWidth
ChannelList.QE.MinInputZPulseWidth

Min. A Pulse width: The minimum pulse width in milliseconds detectable at the A input

Min. B Pulse width: The minimum pulse width in milliseconds detectable at the B input

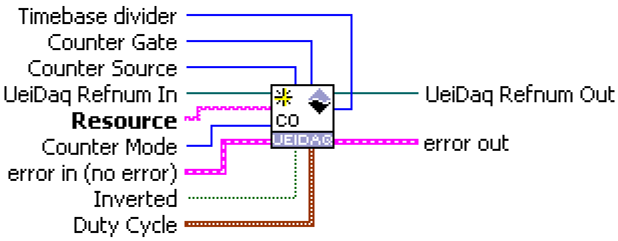
Min. Z Pulse width: The minimum pulse width in milliseconds detectable at the Z input

5.6. *Counter Output*

This session performs single pulse, N pulses or continuous PWM output.

Pulse shape parameters are specified as multiples of the timebase.
For example to output a 1kHz square wave, set low and high tick counts to 66000000/500 = 132000

Use the **Counter Output:Output** instance of **UeiDaqCreateSession.vi**



Counter Source The timebase used to determine the shape of the signal generated on the counter output

Counter Gate The signal that specify whether the counter/timer is on or off

Counter Mode The mode that specify whether the session generate a pulse or a pulse train on the counter output

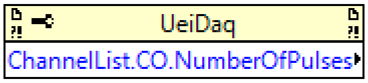
Inverted Specifies whether the signal at the source is inverted before performing the operation

Duty Cycle Specifies how long the output stays low and how long it stays high, the value is in number of ticks of the signal connected at the source

- **Tick1**: the pulse low state duration in number of timebase tick
- **Tick2**: the pulse high state duration in number of timebase tick

Timebase divider The divider used to scale the timebase speed

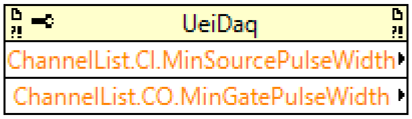
You can set the number of pulses to output using the UeiDaq property node



UeiDaq

ChannelList.CO.NumberOfPulses

Some counter input devices come with digital input filters to reject noisy pulses that are below a programmable width. You can set those filters using the UeiDaq property node:



UeiDaq

ChannelList.CI.MinSourcePulseWidth

ChannelList.CO.MinGatePulseWidth

Min. Source Pulse width: The minimum pulse width in milliseconds detectable at the source input

Min. Gate Pulse width: The minimum pulse width in milliseconds detectable at the gate input

5.7. Messaging

5.7.1. CAN bus

5.7.2. Serial Port

5.7.3. ARINC-429 Input

5.7.4. ARINC-429 Output

5.7.5. MIL-1553 bus

5.8. Miscellaneous

5.8.1. Device Enumeration

5.8.2. Device Reset

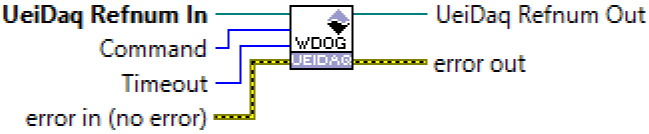
5.8.3. Watchdog

The CPU card on PowerDNA/DNR devices comes with a watchdog timer.
This watchdog is meant to reset the CPU in case the firmware hangs or the network gets disconnected. It does a hard reset on the CPU (immediate reboot).
Outputs will not go to shutdown state

After a WD initiated reset, the outputs will go to init state after a few seconds (once the firmware is started).

Watchdog is only programmable through a session opened on the CPU (which is always device 14).

Use the VI **UeiDaqSetDeviceWatchDogCommand.vi** to control the watchdog timer.



This VI configures the watchdog timer to reset the CPU when a programmable timer delay expires. Periodically clearing the timer prevents reset.

Command: selects what type of events/actions clears the WD timer

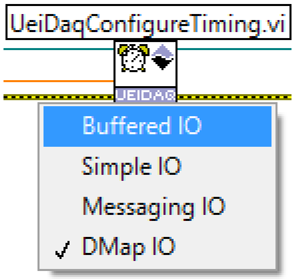
- **Disable**: Disable watchdog
- **EnableClearOnReceive**: Configure watchdog to clear timer upon receiving any packet from host

- **EnableClearOnTransmit**: Configure watchdog to clear timer upon transmitting a packet to the host
- **EnableClearOnCommand**: Configure watchdog to clear timer upon receiving a "manual" clear command
- **ClearTimer**: Manually clear the timer (only works if WD has already been configured with EnableClearOnCommand

Timeout: The timeout delay programmed into the watchdog timer (in milli-seconds)

6. Timing Configuration

The next step after creating the session and specifying the channels is to configure the timing.
The UeiDaq VI library uses the polymorphic VI “UeiDaqConfigureTiming.vi” to let you select one of the three timing modes using the polymorphic selector.



Look at the description of the UeiDaqConfigureTimingFor*** VIs in the Reference section below for detailed information about each timing mode parameter.

6.1. Simple IO:

This mode uses a software clock to time the data acquisition/generation and is well suited for slow speed operations (less than 500Hz). Acquired/Generated data is transferred scans by scans from/to the device.
Data is not buffered; each call to read or write data directly causes hardware access, which can be expensive in CPU time.

6.2. Buffered IO:

This mode uses a hardware clock to time the data acquisition/generation. It is well suited to perform high-speed data acquisition or generation. Acquired/Generated data is transferred in blocks from/to the device.

The buffered timing mode uses internally multiple buffers (called frames) in a mechanism called advanced circular buffer (ACB). Each buffer is represented by a 2D array in LabVIEW.

In the case of an input session, the driver continuously fills buffers with data from the device and sends them to the user application, which can process them at its own pace. For an output session, the user application can fill multiple buffers and send them to the driver without waiting for the device to be ready to accept them.

Having multiple buffer helps avoiding gaps in the acquired or generated data especially under non real-time operating systems such as Windows. Indeed general purpose operating systems sometime takes away CPU cycles from your application to run various background tasks such as services, servers, network requests, virus scanning… During that time the driver keeps filling-up buffers with data that your application will receive as soon as the operating system allows it to run.
Another advantage of circular buffers is when your application takes more time to process a buffer than the time it took to acquire it. It allows your application to later catch up and avoid losing any data. Of course this will only work if this behavior is not recurrent.

The size of each buffer is equal to the buffer size parameter specified when calling "UeiDaqConfigureTimingForBufferedIO.vi", the default number of buffers is 4. You can change the number of buffer using the property node "Data.NumberOfFrames".

The circular buffer uses read and write pointers to keep track of the state of both reader and writers. The write pointer position moves along as the writer is writing new data into the circular buffer and the read pointer does the same as the reader is reading.

In input sessions, the reader is the user program and the writer is the Data Acquisition device.
In output sessions, the reader is the Data Acquisition device and the writer is the user program.

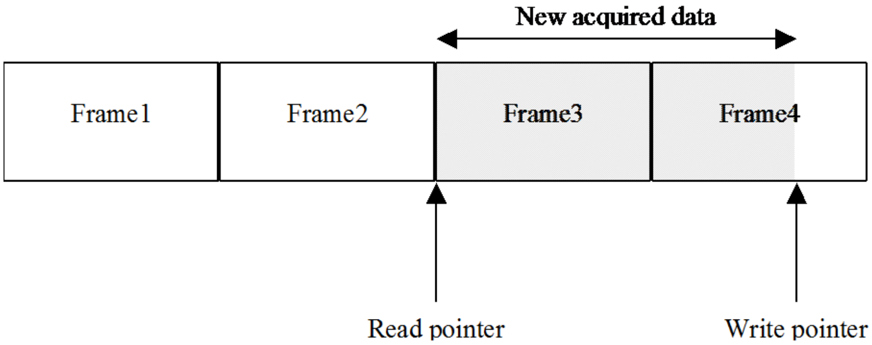


Figure 4:

Figure 4 shows a typical case where the reader is following the writer closely.

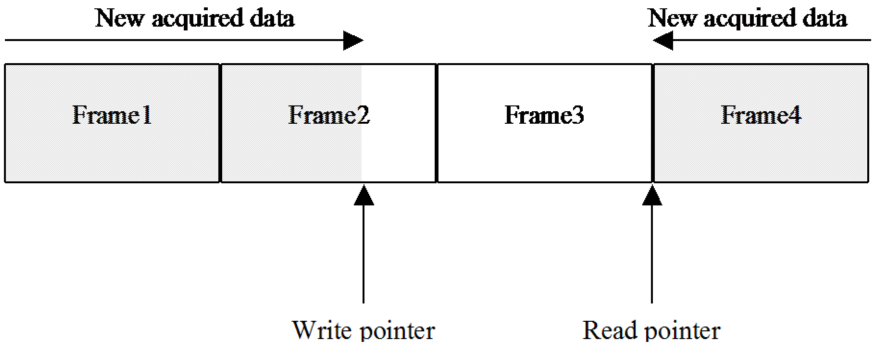


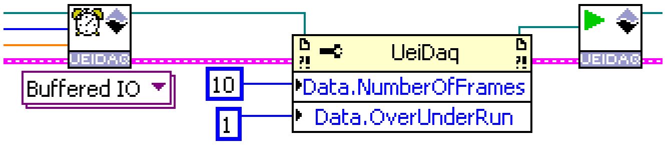
Figure 5:

Figure 5 shows a case where the reader is late and the writer is about to catch up with it. This can happen if your application takes too long to process incoming buffers or doesn't send outgoing buffers fast enough.
A buffer over-run or under-run error will occur when the write pointer meets the reader pointer and the session will abort.

For input applications where losing data doesn't matter you can tell the session to ignore the over-run error and the writer will over-write buffers whose data hasn't been processed yet with new data. The write pointer will just "pass" the read pointer without sending any error to the application.
The same thing is possible for output applications; the reader will recycle already generated data and send it again to the device.

You can disable the over/under run error notification by using the "Data.OverUnderRun" property.

The following sample shows how to set the number of frames and disable the buffer over-run error.

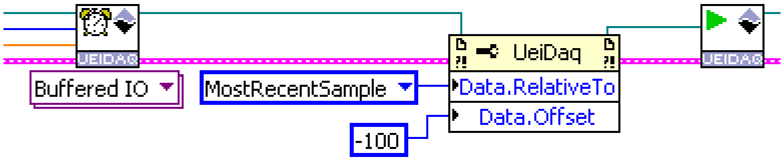


By default, the reader reads the oldest data from the circular buffer and updates the read pointer accordingly. However you can arbitrarily set the position of the read pointer by using the properties "Data.RelativeTo" and "Data.Offset".
This can be useful to skip unread data when you know your application is getting late or to always read the most recently acquired data and discard any "older" data.

You can set the "Data.RelativeTo" property to "CurrentPosition" (its default value) or "MostRecentSample" which correspond to the position of the write pointer.
The "Data.Offset" property specifies the new position of the read pointer relative to the "Data.RelativeTo" property. Its value can be negative to move the read pointer backward or positive to move it forward.

For example to immediately read the most recently acquired 100 samples you would set "Data.RelativeTo" to "MostRecentSample" and "Data.Offset" to –100. This will move the read pointer 100 scans before the write pointer.

The following sample shows how to change the read pointer position.



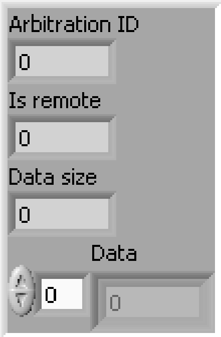
You can access the number of scans that are available to be read from (or written to) the circular buffer using the property "Data.AvailableScans".
You can also get the total number of scans read from (or written to) the circular buffer using the property "Data.TotalScans".

6.3. Messaging IO:

This mode is used with message communication devices such as serial ports and CAN bus interfaces. What defines a message depends on the communication port type.

On serial ports, messages are simply bytes, represented by LabVIEW string data type.

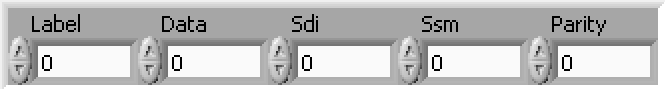
On CAN ports messages are CAN frames represented by the cluster below.



Arbitration ID
0
Is remote
0
Data size
0
Data
0 0

- Arbitration ID: CAN Frame arbitration id
- Is remote: Specifies whether this is a remote frame
- Data Size: The number of significant bytes in the payload
- Data: The frame's payload. It can contain up to 8 bytes

On ARINC-429 ports, messages are ARINC word represented by the cluster below.



Label	Data	Sdi	Ssm	Parity
0	0	0	0	0

- Label: The label of the word. It is used to determine the data type of the Data field, therefore, the method of data translation to use.
- Ssm: Sign/Status Matrix or SSM. This field contains hardware equipment condition, operational mode, or validity of data content.
- Sdi: Source/Destination Identifier or SDI. This is used for multiple receivers to identify the receiver for which the data is destined.
- Parity: The parity bit.
- Data: The payload of the word. Its format depends on the label. Most common formats are BCD (binary-coded-decimal) encoding, BNR (binary) encoding or discrete format where each bit represents a Pass/Fail, True/False or Activated/Non-Activated condition.

6.4. DMAP IO

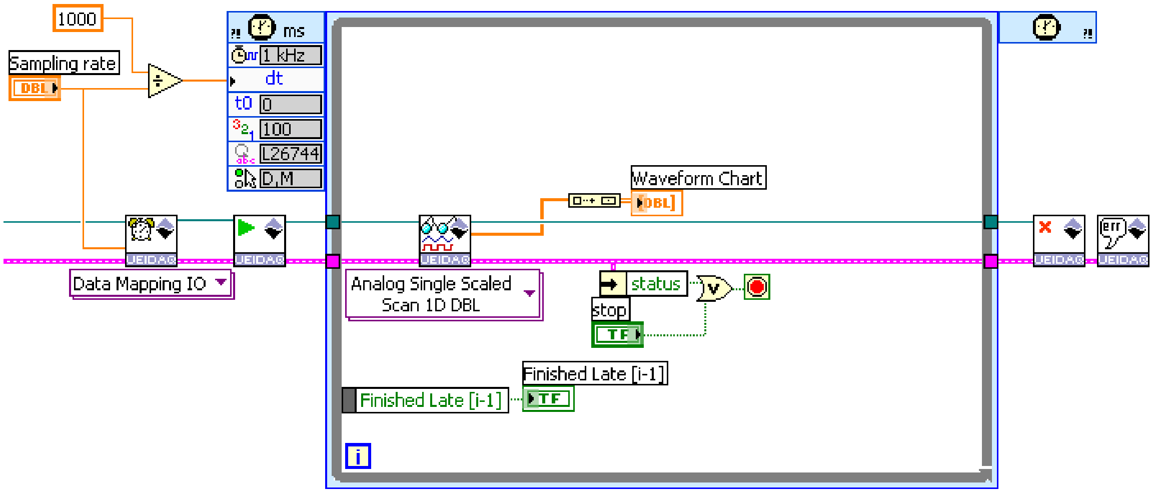
The “data map” timing mode (also called DMAP mode) is only available with PowerDNA devices. It allows for transfer of single scans at a given rate timed by a hardware clock.
This mode is very useful for real-time applications that need to acquire and process scans one by one but at a fixed rate.

This mode offers better performances than the “point by point” timing mode where each device is polled one by one in a software timed loop. In contrast, using the DMAP timing mode, all PowerDNA devices within one IO module are read simultaneously and the resulting data is transferred from the PowerDNA IO module to the host in one operation.

The data transfer operates in the background at a specified rate so that your application never has to wait for the transfer to take place

The acquired data is returned as an array containing one sample per channel.
The data to be generated must be passed as an array containing one value for each channel.

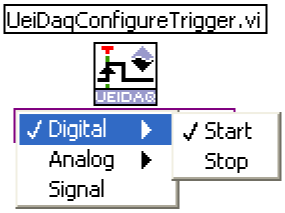
The best way to take advantage of this mode under LabVIEW is to use a timed loop programmed to run at the same rate than the DMAP.



7. Trigger Configuration

The session automatically configures triggers by default to immediately start and stop operation on software request. You can override this and configure the session to start and or stop when an external event occurs.

The UeiDaq VI library uses the polymorphic VI "UeiDaqConfigureTrigger.vi" to let you select one of the trigger modes using the polymorphic selector.

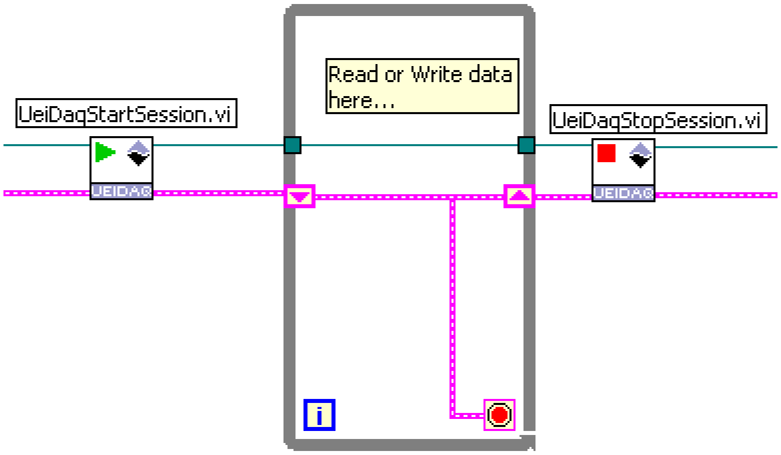


- Digital: Starts or stops the session when a digital edge is applied on the external trigger input of the device.
- Analog: Starts the acquisition in the background and waits for the configured condition (level and hysteresis) to be met before returning any data.
- Signal: Starts the session on the occurrence of an external signal such as a backplane synchronization line available with PXI and PowerDNA devices.

Look at the description of the UeiDaqConfigure**Trigger VIs in the Reference section below for detailed information about their parameters.

8. Session Control

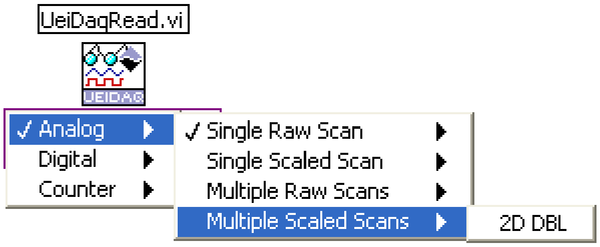
You can start and stop the session by calling the VIs "UeiDaqStartSession.vi" and "UeiDaqStopSession.vi".



Note that if you don't explicitly start the session, it will be automatically started the first time you try to transfer data.

9. Data Transfer

The UeiDaq VI library uses the polymorphic VIs "UeiDaqRead.vi" and "UeiDaqWrite.vi" VIs to transfer data from/to the device.
You can select the type of data to transfer using the polymorphic VI selector.



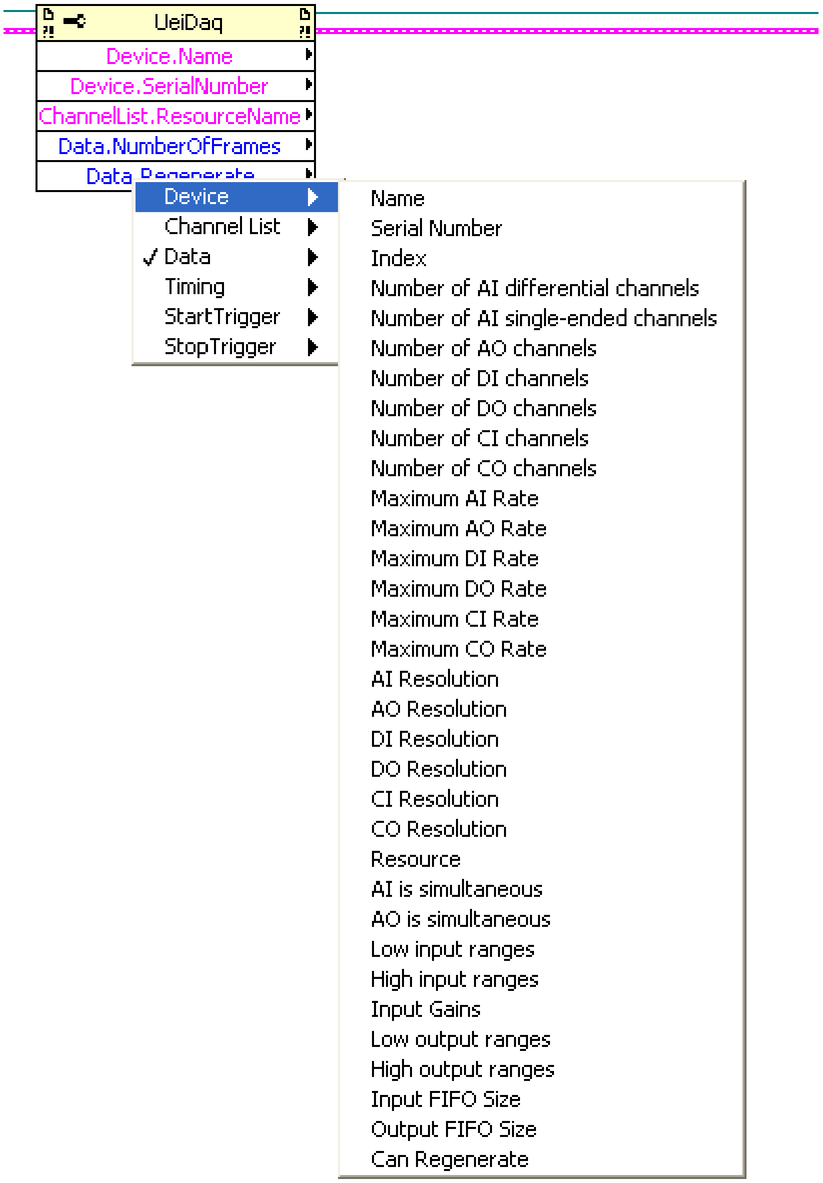
- Single Scan: Transfer data from/to the device one scan (one value per channel) at a time.
- Multiple Scans: Transfer multiple scans from/to the device. This operation will block until the specified amount of data is ready to be retrieved/accepted from/by the device.

With analog devices, you can specify if you wish to transfer the data as raw values or a scaled measurement values.

With digital and counter/timer devices you can specify whether the data is to be transferred using 16-bit or 32-bit integers. Read the specification of your device to determine which one you should use.

10. Property Node

The UeiDaq VI library VIs give you access to the most commonly used session settings from their connector pane. You can read back those settings and configure other advanced settings using LabVIEW’s property node.



Look at the appendix “Session Properties” for an exhaustive list and description of each property.

11. Error Handling

The UeiDaq VI library comes with its own range of error codes. You can use LabVIEW's standard error handler VI or the VI "UeiDaqShowError.vi" to translate UeiDaq error codes to a human readable error message.

Look at the appendix "Error Codes" for an exhaustive list and description of each error.

Appendix A: Error Codes

C/C++ Constant	*Code*	*Description*
UEIDAQ_SUCCESS	0	Success
UEIDAQ_ATTRIBUTE_INVALID_ERROR	0x80290001	The specified attribute doesn't exist
UEIDAQ_ATTRIBUTE_BAD_TYPE_ERROR	0x80290002	The specified attribute exists but the type is wrong
UEIDAQ_TIMEOUT_ERROR	0x80290003	Timeout occurred
UEIDAQ_BAD_PARAMETER_ERROR	0x80290004	One of the specified parameter(s) is invalid
UEIDAQ_DEVICE_INVALID_ERROR	0x80290005	The specified device doesn't exist
UEIDAQ_CHANNEL_INVALID_ERROR	0x80290006	The specified channel doesn't exist
UEIDAQ_ATTRIBUTE_OUT_OF_RANGE_ERROR	0x80290007	The specified attribute's value is out of range
UEIDAQ_DEVICE_CAPABILITY_ERROR	0x80290008	The device is not capable of such an operation
UEIDAQ_BAD_RESOURCE_STRING_ERROR	0x80290009	The resource string is incorrectly formatted
UEIDAQ_DRIVER_INVALID_ERROR	0x8029000A	The specified driver doesn't exist
UEIDAQ_SESSION_INVALID_ERROR	0x8029000B	The session is invalid
UEIDAQ_SUBSYSTEM_ALREADY_USED_ERROR	0x8029000C	The subsystem is already reserved by another session
UEIDAQ_INVALID_STATE_ERROR	0x8029000D	The session is not in a state allowing this operation
UEIDAQ_BUFFER_OVERRUN_ERROR	0x8029000E	Data was lost because it was not read fast enough
UEIDAQ_BUFFER_UNDERRUN_ERROR	0x8029000F	Not enough data was available to keep the generation running
UEIDAQ_SUBSYSTEM_BUSY_ERROR	0x80290010	The subsystem associated with this session is already used by another session
UEIDAQ_SUBSYSTEM_INTERNAL_ERROR	0x80290011	An unknown error occurred in the plugin
UEIDAQ_BUFFER_ERROR	0x80290012	An error occurred while reading/writing the buffer
UEIDAQ_DEVICE_ERROR	0x80290013	An error occurred while accessing the device
UEIDAQ_SUBSYSTEM_NOT_BUFFERED_ERROR	0x80290014	This subsystem doesn't support buffered IO

UEIDAQ_NOT_ENOUGH_MEMORY_ERROR	0x80290015	There was not enough memory to complete this operation
UEIDAQ_WRONG_DATA_TYPE_ERROR	0x80290016	The specified data type doesn't match the device capability
UEIDAQ_NO_MORE_ITEMS_ERROR	0x80290017	There is no more data available
UEIDAQ_OPERATION_INVALID_ERROR	0x80290018	The operation is incompatible with the current session
UEIDAQ_NOT_ENOUGH_DATA_ERROR	0x80290019	There were not enough data passed to complete the operation
UEIDAQ_INVALID_TC_TYPE_ERROR	0x8029001A	The thermocouple type is invalid
UEIDAQ_INVALID_TEMP_SCALE_ERROR	0x8029001B	The temperature scale is invalid
UEIDAQ_INVALID_BRIDGE_CONFIG_ERROR	0x8029001C	The strain gauge bridge configuration is invalid
UEIDAQ_INVALID_RTD_CONV_ERROR	0x8029001D	The RTD conversion is invalid
UEIDAQ_DIVIDE_BY_ZERO_ERROR	0x8029001E	Division by zero
UEIDAQ_CONVERSION_ERROR	0x8029001F	An error happened while converting units
UEIDAQ_INVALID_RATE_ERROR	0x80290020	The specified rate is above the device maximum rate or violates the settling time
UEIDAQ_CTR_TMR_MODE_INVALID_ERROR	0x80290021	The counter/timer do not support the specified mode
UEIDAQ_CTR_TMR_IN_USE_ERROR	0x80290022	The counter/timer is already used by another session
UEIDAQ_CAN_NOT_REGENERATE_ERROR	0x80290023	The device doesn't support regeneration
UEIDAQ_CHANNEL_LIST_POWER_OF_2_ERROR	0x80290024	The Channel list size must be a power of 2
UEIDAQ_SESSION_XML_ERROR	0x80290025	The XML session string is invalid
UEIDAQ_DIGITAL_PORT_IN_USE_ERROR	0x80290026	The digital port is already used by another session
UEIDAQ_INVALID_TIMING_MODE_ERROR	0x80290027	The device associated with the session doesn't support the specified timing mode
UEIDAQ_ASYNC_IN_PROGRESS_ERROR	0x80290028	An asynchronous operation is already in progress
UEIDAQ_NOT_IMPLEMENTED_ERROR	0x80290029	The feature is not yet implemented
UEIDAQ_CONFIG_INVALID_ERROR	0x8029002A	The current configuration settings were rejected by the device

`UEIDAQ_DEVICE_NOT_RESPONDING_ERROR`	0x8029002B	The device is not responding, check the connection and the device's status
`UEIDAQ_INVALID_TRIGGER_SIGNAL_ERROR`	0x8029002C	The specified trigger signal is not compatible with the device associated to this session
`UEIDAQ_SIGNAL_BUSY_ERROR`	0x8029002D	The specified trigger or clock can't be routed to the device

Appendix B: Properties

Property Name	Description
Device.Name	The name of the device associated with this session
Device.SerialNumber	The serial number of the device associated with this session
Device.Index	The index of the device associated with this session
Device.NumberOfAIDiffChannels	The number of Analog input differential channels on the device associated with this session
Device.NumberOfAISEChannels	The number of Analog input single-ended channels on the device associated with this session
Device.NumberOfAOChannels	The number of Analog output channels on the device associated with this session
Device.NumberOfDIChannels	The number of Digital input channels on the device associated with this session
Device.NumberOfDOChannels	The number of Digital output channels on the device associated with this session
Device.NumberOfCIChannels	The number of Counter input channels on the device associated with this session
Device.NumberOfCOChannels	The number of Counter output channels on the device associated with this session
Device.AIRate	The maximum Analog input rate of the device associated with this session
Device.AORate	The maximum Analog output rate of the device associated with this session
Device.DIRate	The maximum Digital input rate of the device associated with this session
Device.DORate	The maximum Digital output rate of the device associated with this session
Device.CIRate	The maximum Counter input rate of the device associated with this session
Device.CORate	The maximum Counter output rate of the device associated with this session
Device.AIResolution	The Analog input resolution of the device associated with this session
Device.AOResolution	The Analog output resolution of the device

	associated with this session
Device.DIResolution	The Digital input resolution of the device associated with this session
Device.DOResolution	The Digital output resolution of the device associated with this session
Device.CIResolution	The Counter input resolution of the device associated with this session
Device.COResolution	The Counter output resolution of the device associated with this session
Device.Resource	The resource string of the device associated with this session
Device.AISimultaneous	The analog inputs are simultaneous
Device.AOSimultaneous	The analog outputs are simultaneous
Device.LoRanges	The array containing the low input ranges
Device.HiRanges	The array containing the high input ranges
Device.Gains	The array containing the input gains
Device.LoAORanges	The array containing the low output ranges
Device.HiAORanges	The array containing the high output ranges
Device.InputFIFOSize	The number of samples that can be stored in the input FIFO
Device.OutputFIFOSize	The number of samples that can be stored in the output FIFO
Device.CanRegenerate	"Specifies whether the device can do output regeneration
ChannelList.ResourceName	The resource strings of the channels in the channel list
ChannelList.AliasName	The alias name of the channels in the channel list
ChannelList.Index	The index of the channels in the channel list
ChannelList.AI.Minimum	The minimum range of the channels in the channel list
ChannelList.AI.Maximum	The maximum range of the channels in the channel list
ChannelList.AI.Gain	The programmed gains of the channels in the channel list
Channel.AI.InputMode	The input mode of the channels in the channel list
ChannelList.TC.CJC	The temperature measured by the cold-junction compensation sensor
ChannelList.AIVEx.Excitation	The voltage level of the excitation

ChannelList.AIVEx.EnableShuntCal	Enable/Disable the shunt calibration circuitry
ChannelList.AIVEx.ShuntLocation	The location of the shunted branch of the wheatstone bridge
ChannelList.AIVEx.ShuntResistance	Resistance programmed to the shunt calibration resistor
ChannelList.AIVEx.GainAdjustmentFactor	The gain adjustment factor used to scale the measurements
Channel.DI.EdgeMask	The edge mask for detecting change detection on the input channels
Channel.CI.CounterSource	The signal connected to the source of the counter input channels
Channel.CI.CounterMode	The operation mode of the counter input channels
Channel.CI.CounterGate	The source of the gate
Channel.CI.Inverted	The inverted flag
Channel.CI.Divisor	The divisor used to scale the timebase
Channel.CI.MinSourcePulseWidth	The minimum pulse width in ms the counter recognizes at its source
Channel.CI.MinGatePulseWidth	The minimum pulse width in ms the counter recognizes at its gate
Channel.CO.CounterSource	The signal connected to the source of the counter output channels
Channel.CO.CounterMode	The operation mode of the counter output channels
Channel.CO.CounterGate	The source of the gate
Channel.CO.Inverted	The inverted flag
Channel.CO.Divisor	The divisor used to scale the timebase
Channel.CO.Tick1	The number of clock ticks while the pulse is low
Channel.CO.Tick2	The number of clock ticks while the pulse is high
Channel.CO.MinSourcePulseWidth	The minimum pulse width in ms the counter recognizes at its source
Channel.CO.MinGatePulseWidth	The minimum pulse width in ms the counter recognizes at its gate
Data.NumberOfScans	The number of scans to read at once
Data.ScanSize	The number of samples per scan (equivalent to the size of the channel list)
Data.SampleSize	The number of bytes per sample (it is a multiple fo 2 bytes)

Data.NumberOfFrames	The number of frames used by the driver in the circular buffer
Data.Regenerate	Specifies whether data will be automatically regenerated from the FIFO (Analog output only)
Data.OverUnderRun	Specifies whether data will be automatically overwritten when it is not read fast enough
Data.CurrentScan	The position of the last acquired scan in the circular buffer
Data.AvailableScans	The number of scans available in the circular buffer
Data.TotalScans	The total number of scans acquired since the session was started
Data.Offset	The offset in the circular where the next read/write operation will start
Data.RelativeTo	Specifies whether the new offset is relative to the last position of the beginning of the circular buffer
Timing.Mode	The timing mode of the session
Timing.ConvertClockSource	The Source of the convert clock
Timing.ScanClockSource	The Source of the scan clock
Timing.ConvertRate	The frequency of the convert clock
Timing.ScanRate	The frequency of the scan clock
Timing.ConvertEdge	The active edge of the convert clock
Timing.ScanEdge	The active edge of the scan clock
Timing.ConvertSignal	The name of the signal used as the conversion clock
Timing.ScanSignal	The name of the signal used as the scan clock
Timing.Duration	Specifies whether the session runs continuously or until the required amount of data si transferred
Timing.Timeout	The maximum time allowed for the data reading/writing operation to complete
Timing.ScanClockDivisor	The divisor used to scale an external scan clock
Timing.ConvertClockDivisor	The divisor used to scale an external convert clock
Timing.ScanClockDividingCounter	The counter used to scale an external scan clock

Timing.ConvertClockDividingCounter	The counter used to scale an external convert clock
StartTrigger.Source	The source of the trigger that will start the session
StartTrigger.Edge	The active edge of the external start trigger
StartTrigger.Action	The action to execute when the trigger occur
StartTrigger.Condition	The condition for the analog software trigger
StartTrigger.Channel	The channel to monitor for the trigger condition
StartTrigger.Level	The scaled value at which the trigger occurs. The value is in the same unit as the measurement
StartTrigger.Hysteresis	The hysteresis window scaled value. The hysteresis window is a noise filter. The trigger occurs when the signal leave the window
StartTrigger.Signal	The name of the signal used to transmit the start trigger
StopTrigger.Source	The source of the trigger that will stop the session
StopTrigger.Edge	The active edge of the external stop trigger
StopTrigger.Action	The action to execute when the trigger occur
StoptTrigger.Condition	The condition for the analog software trigger
StopTrigger.Channel	The channel to monitor for the trigger condition
StopTrigger.Level	The scaled value at which the trigger occurs. The value is in the same unit as the measurement
StopTrigger.Hysteresis	The hysteresis window scaled value. The hysteresis window is a noise filter. The trigger occurs when the signal leave the window
StopTrigger.Signal	The name of the signal used to transmit the stop trigger