

COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

 Sell For Cash  Get Credit  Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New**, **New Surplus**, **Refurbished**, and **Reconditioned** NI Hardware.



Bridging the gap between the manufacturer and your legacy test system.

 1-800-915-6216

 www.apexwaves.com

 sales@apexwaves.com

All trademarks, brands, and brand names are the property of their respective owners.

Request a Quote

 **CLICK HERE**

NI-9853

DRIVVEN

NO_x Sensor Kit User's Manual

cRIO Module Platform

D00018 Rev A

D00019 Rev A

August 2010



Contents

Introduction	3
System Diagram	4
Hardware	5
Powering the Hardware	5
Continental Smart NO _x Sensor Module Specifications.....	6
cRIO Platform Compatibility	9
Software.....	10
Creating a LabVIEW Project.....	11
Exploring the cRIO Example Project	13
cRIO Sub VI Documentation	18
Wiring Harness Documentation.....	22

Introduction

The Driven cRIO and USB NO_x Sensor Module Kits provides an interface for one or two NO_x and O₂ exhaust gas sensors.

Features:

- 1-Ch. or 2 Ch. system
- Two signals output:
 - NO_x concentration (ppm)
 - O₂ concentration (%)
- Sensor controller supply voltage of 12V to 32V
- Reverse battery protection on sensor controller module
- Wide range of applications:
 - Gasoline / Diesel
 - Passenger car / Truck
- Helps to achieve emissions requirements
- Includes sensor controller module harness
- Multiple platform support including USB and cRIO
- cRIO module based platform provides
 - LabVIEW FPGA and RT VIs included for quick integration with application
 - Integration with existing PXI or cRIO chassis hardware

System Diagram

Driven cRIO NO_x Sensor Module Kit

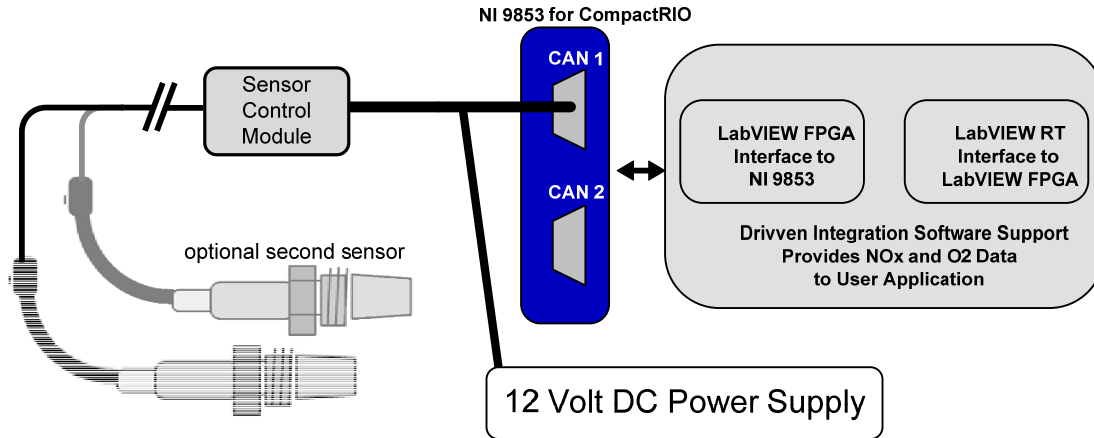


Figure 1 cRIO System Drawing

Hardware

The cRIO based kit provides the following hardware:

- 1** Continental Smart NO_x Sensor with integrated sensor controller module
- 1** Sensor bung
- 1 Connector kit for NO_x sensor controller module
- 1 National Instruments NI-9853 CompactRIO CAN module
- **quantity of 2 if part # D000019

The Drivven NO_x software toolkit is only compatible with the Continental Uninox Smart NO_x Sensor having a part number of 5WK96622A. There are many additional Continental NO_x sensor modules, found in various OEM engine applications, which communicate over CAN. They likely have a different part number, which means they will not be compatible with this kit because each of the OEM sensors are programmed with proprietary communication protocols. Drivven cannot support those sensors. The NO_x sensor modules provided by Drivven are engineering samples procured directly from Continental. If you already have the correct sensor module, then you may purchase the NO_x sensor module kit without the included sensor module. Please contact Drivven for a quote.

The wiring harness that is included with the USB NO_x kit is a five foot cable that connects to the NO_x Sensor Control Module and then splits into two cables for the NI-9853 module and an external power supply. The first cable connects to the NI-9853 with a female DB-9 connector. A terminating resistor of approximately 120 ohms is hidden inside this cable near the NO_x Sensor Control Module, between the CAN High and CAN Low wires. The second cable provides three un-terminated leads for connecting to a power supply (2 leads) and for address selection of the NO_x sensor module (1 lead). If the cable is to be extended, CAN network wiring guidelines must be followed. Please refer to the operating instructions provided with the NI-9853 CAN module for more details.

Powering the Hardware

The NO_x Sensor Controller Module requires power from a range of 12V to 16V with a continuous current of 0.6A and a peak current of 12A. The maximum power requirement is 20W.

For power requirements of the NI 9853, please refer to the National Instruments operating instructions provided with the hardware.

Continental Smart NO_x Sensor Module Specifications

Table 2. NO_x Sensor Performance Specifications

Output Type	Measurement Range	Accuracy	Response Time (33-66%)	Data Update Rate
NO _x	0 – 3000 ppm	@ 0 ppm: ±10 ppm @ 100 – 1500 ppm: ±10% @ 1500 – 3000 ppm: undefined	1300 ms (fresh) 1650 ms (aged)	50 ms interval @ 250 kBaud
O ₂	-12 – 21 [%]	@ λ=0.90: ±1.4% (fresh) @ 0% (λ=1.00): ±0.13% (fresh) @ 0% (λ=1.00): ±0.25% (aged) @ 13% (λ=2.65): ±0.32% (fresh)	1000 ms (fresh) 1300 ms (aged)	

NO_x Sensor Light-off times (Conditions: Air T = 25 ±5°C, BattV = 28V, Heater turned ON)

NO_x < 100 sec
O₂ < 80 sec

NO_x Sensor Preheating Function

If power supply is on, the sensor is in preheating mode until the **Heater** Boolean is set to TRUE. If the **Heater** Boolean is set to FALSE, the sensor returns to preheating mode. The preheating mode protects the sensor from mechanical cracks caused by water splash.

NO_x Sensor Operating Temperature Ranges

Minimum sensor module controller temperature -40°C
Maximum sensor module controller temperature 105°C
Sensor module controller temperature range of 85°C to 100°C allowed for up to 10 minutes

Minimum storage temperature -40°C
Maximum storage temperature 120°C
Maximum storage time 2 years

Maximum exhaust gas temperature 800°C
Exhaust gas temperature of 950°C allowed up to 100 hours

Maximum sensor hexagon screw temperature 620°C
Sensor hexagon screw temperature of 650°C allowed for up to 100 hours

Maximum sensor grommet temperature 200°C
Sensor grommet temperature of 230°C allowed for up to 100 hours

Minimum preheating sensor temperature 80°C
Maximum preheating sensor temperature 120°C

Lifespan approved by life cycle pattern 2000 hours or 120K miles

NO_x Sensor Electrical Characteristics

NO_x Sensor Supply Voltage

Minimum supply voltage 12 V
Maximum supply voltage 16V
If supply voltage > 32 V, sensor is not operated

NO_x Sensor Supply Current

Average supply current	0.6 A
Peak supply current at switch on	12 A
Inrush current	20 A

Supply Power

Maximum supply power	20 W
----------------------	------

NO_x Sensor Misc.

Thread Torque	50 Nm
---------------	-------

Lubrication	Anti-Seize Compound
-------------	---------------------

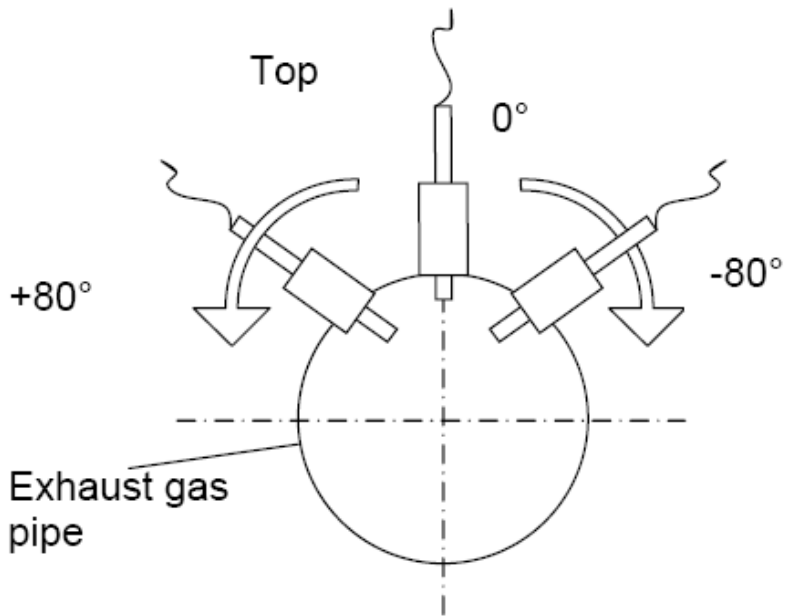


Figure 2 Installation Position

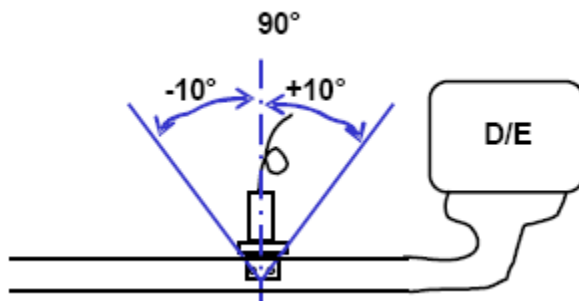


Figure 3 Tilt Angle in Gas Flow Direction

NO_x Sensor Controller Module Connector

Type of connector

Number of pins

Connector pin assignment

Hirschmann MLK 872-860-501

5

Pin 1: Battery

Pin 2: Ground

Pin 3: CAN Low

Pin 4: CAN High

Pin 5: Address Switch

Pulling pin 5 to ground changes the CAN arbitrary ID of the sensor control module so that two NO_x sensor control modules can be added to the same network.

cRIO Platform Compatibility

CompactRIO modules from National Instruments are compatible within two different platforms from National Instruments. One platform is CompactRIO, consisting of a CompactRIO controller and CompactRIO chassis as shown in Figure 1a below.



Figure 4 CompactRIO platform compatible with Driven/NI CompactRIO modules

The other platform is National Instruments PXI which consists of any National Instruments PXI chassis along with a PXI RT controller and PXI-78xxR R-Series FPGA card. An R-Series expansion chassis must be connected to the PXI FPGA card via a SHC68-68-RDIO cable. The CompactRIO modules insert into the R-Series expansion chassis. This platform is shown in Figure 1b below.

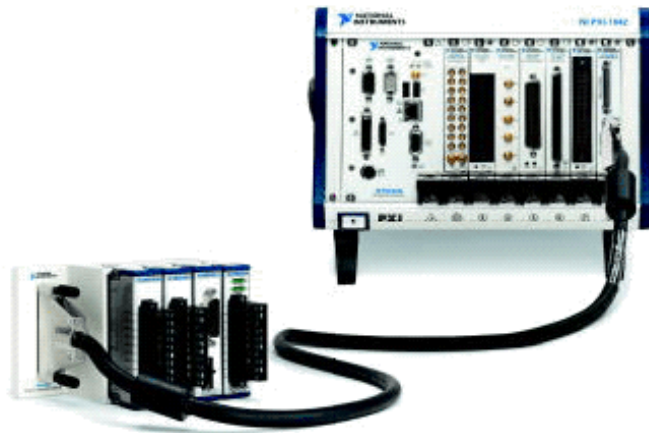


Figure 5 PXI platform compatible with Driven/NI CompactRIO modules

National Instruments NI 9853 CompactRIO modules are not compatible with the National Instruments CompactDAQ chassis.

Driven CompactRIO modules REQUIRE one of the hardware support systems described above in order to function. The modules may not be used by themselves and/or interfaced to third party devices at the backplane HD15 connector. These efforts will not be supported by Driven or National Instruments.

Software

The NO_x Sensor cRIO Module Kit is provided with an installer package which may be downloaded from Drivven’s Sharepoint website after obtaining login access from Drivven. User’s may go to <http://portal.drivven.com/SoftwareDownload> and enter the provided username and password to gain access to the specific product installer packages which have been purchased. The installer packages are executables which should be run on the intended development computer, having LabVIEW development tools installed. After installing the package, a “Start->Programs->Drivven->ProductRelease” menu item will be added to the desktop. The specific product will have an example LabVIEW project appear under the “Examples” menu and the user manual will appear under the “Manuals” menu. User’s may copy and open the example project to experiment with the module or use as a starting point for a new application. All software files, example projects and documentation are installed to:

C:\Program Files\National Instruments\LabVIEW X.X\vi.lib\addons\DrivvenProductRelease\.

When working with block diagrams, user’s will notice a “Drivven” function palette added to the standard LabVIEW palette, specific for the RT or FPGA target. VIs for a specific Drivven product will be categorized according to product name. Also, some Drivven products will install RT and FPGA VIs under a “General” function palette which is intended to be used across multiple products.

Requirements

The Drivven cRIO based VIs require:

- LabVIEW 8.6 Full Development or later
- LabVIEW RT Module 8.6 or later
- LabVIEW FPGA Module 8.6 or later
- NI-RIO 2.4 or later

The NO_x Sensor cRIO Module Kit is provided with both a LabVIEW FPGA VI for interfacing to the NI cRIO 9853 CAN module and a LabVIEW RT VI for interfacing with the FPGA VI and managing and reporting sensor results.

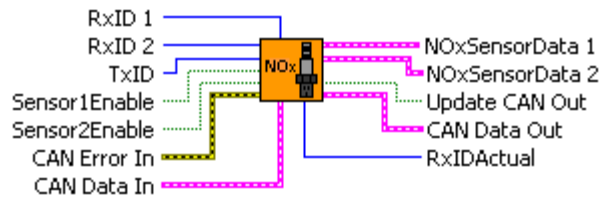


Figure 6 nox_revx.vi and nox_rt_revx.vi icons with leads

Creating a LabVIEW Project

Drivven recommends working from the provided example application as a starting point for learning the use of the Drivven software blocks. However, the following section describes starting a LabVIEW project from scratch and adding a CompactRIO module. For the USB based platform the LabVIEW example project is not required for operation but it does provide a useful means of viewing the applicable VIs. The following directions are not applicable for the USB platform because there is no need to use a real time target when only using the NI-8473.

- 1.) Install the Drivven software by running the installer executable and accepting the software license agreement.
- 2.) Restart LabVIEW, if previously running, and create a new LabVIEW project.
- 3.) Give the new project a name by clicking the “Save Project” button on the project toolbar.
- 4.) Right click on the highest item in the project hierarchy (“Project:...”) and navigate to “New->Targets and Devices...”
- 5.) Within the “Add Targets and Devices...” dialog, select the appropriate radio button, depending on whether you already have an existing powered and configured RT target on the network or if you are adding a new RT target which is not present yet on the network.
 - a. Existing Target or Device
 - i. Expand the appropriate category in the “Targets and Devices” list to see the discovered targets in that category.
 - ii. Double-click the desired target to add it to your project.
 - b. New Target or Device
 - i. Expand the appropriate category in the “Targets and Devices” list to see all possible targets within that category.
 - ii. Double-click the desired target to add it to your project.
- 6.) If the new RT target is not currently on the network, right-click on the RT target within the project and open the properties dialog to set the IP address or DNS name if necessary.
- 7.) Right-click on the RT target within the project and navigate to “New->Targets and Devices...”
- 8.) Within the “Add Targets and Devices...” dialog, select the appropriate radio button, depending on whether you already have an existing FPGA target connected to an existing RT target or if you are adding a new FPGA target which is not present yet.
 - a. Existing Target or Device
 - i. Expand the appropriate category in the “Targets and Devices” list to see the discovered FPGA targets in that category.
 - ii. Double-click the desired target to add it to your project.
 - b. New Target or Device
 - i. Expand the appropriate category in the “Targets and Devices” list to see all possible targets within that category
 - ii. Double-click the desired target to add it to your project.
 - c. If working with a CompactRIO RT target, then an appropriate CompactRIO chassis must be added before adding an FPGA target. An FPGA target must be added to the cRIO chassis in order to force LabVIEW FPGA Interface mode. Scan Interface mode is not appropriate for working with Drivven products. The mode can also be set by right-clicking on the cRIO chassis within your project and opening the properties dialog.
- 9.) After adding the FPGA target, right-click on the FPGA target within the project and open the properties dialog to set the resource name if not set correctly already. The resource name can be found from MAX when connected to the actual remote system.
- 10.) If the FPGA target is a PXI or PCI card, then an R Series expansion chassis must be added under the FPGA target. This is done by right-clicking on the FPGA target and

- navigating to “New->R Series Expansion Chassis”. Within the following dialog, select the appropriate FPGA connector to which the chassis will be connected. A unique name for the chassis may also be specified.
- 11.) Right click on the R-Series expansion chassis or cRIO FPGA target and navigate to “New->C Series Modules...”
 - 12.) Select the “New Target or Device” radio button and double-click on the “C Series Module” in the “Targets and Devices” list. In the following dialog, select the desired Drivven module at the bottom of the “Module Type” list. The Drivven modules will be appended there if any Drivven module software has been installed. Select the appropriate module location. Finally, specify an appropriate name for the module, which will later appear in the FPGA I/O nodes in the FPGA block diagram. Having meaningful module names is important for preventing coding mistakes.
 - 13.) After adding a module to the project, a folder will automatically be added to the project having the same module name given in the module configuration dialog. The folder will contain the FPGA I/O pins for the module slot. These I/O pins can be selected in the block diagram when connecting the module VI PinInput and PinOutput clusters to FPGA I/O nodes. The example application, discussed below, should be consulted for further details about connecting the PinInput and PinOutput clusters to FPGA I/O nodes. Within the example projects, notice the FPGA I/O node elements having module name prefixes.
 - 14.) Some Drivven modules can be automatically recognized by LabVIEW when adding cRIO modules to the project. However, Drivven does not recommend using this feature because the module names, which are automatically assigned, are not meaningful (Mod1, Mod2, etc) and can lead to coding mistakes when wiring the Drivven FPGA VIs to the I/O nodes. Adding the modules to the project manually, as described above, is still the recommended method.

Exploring the cRIO Example Project

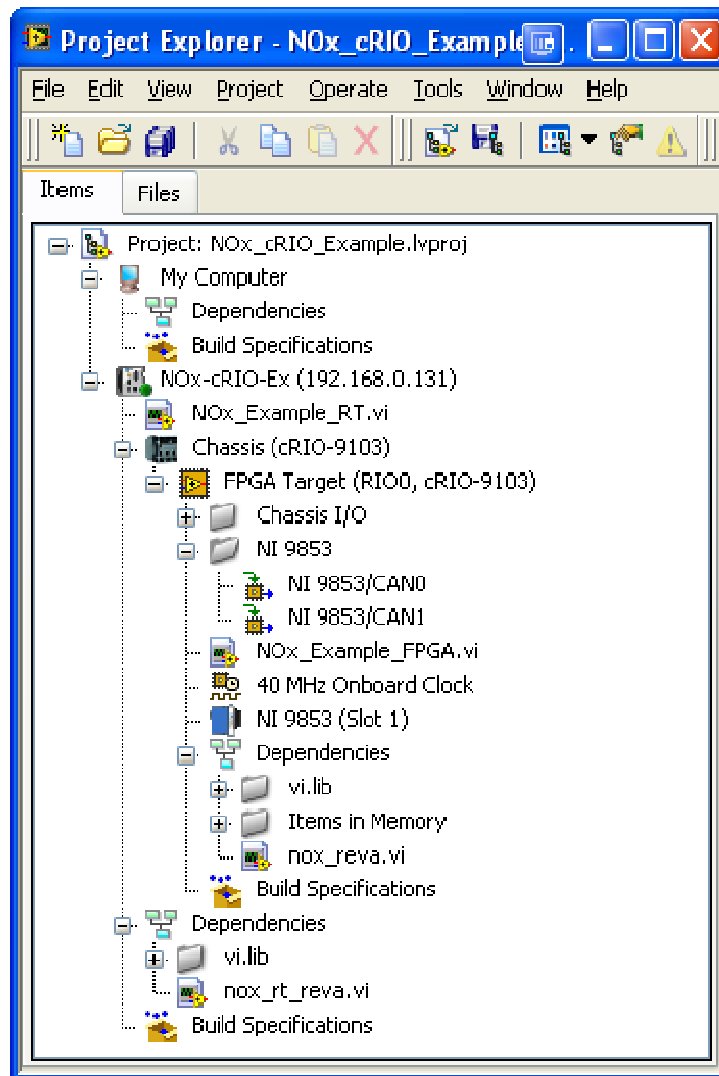


Figure 7 Example expanded project explorer of NO_x_Example.lvproj

Use the NO_x_Example_FPGA.vi as an example to implement a NO_x sensor interface into your own application. Just copy and paste one of the while loops, as shown below in figure 4, into your application.

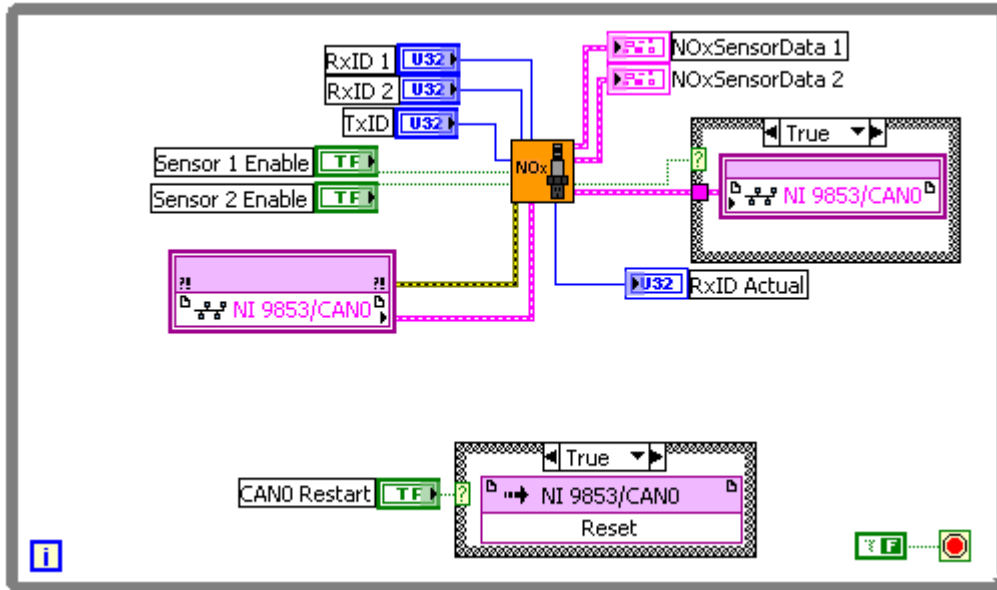


Figure 8 Example block diagram implementation of NO_x VI

Please follow the guidelines below for adding the nox_reva.vi to your FPGA application:

- 1.) The NO_x sensor interface vi, nox_reva.vi, should be placed in its own while loop.
- 2.) Do not place more than one NO_x sensor interface vi within the same while loop.
- 3.) Do not place the NO_x sensor interface within a single cycle loop.
- 4.) The cRIO 9853 CAN Module properties may be configured by right-clicking on the module within the project explorer. The properties should be set to the following:
 - a. Baud Rate = 250 Kbps
 - b. Auto Start = Checked
 - c. Listen Only = Unchecked
 - d. Input Timeout > 50 msec (100 msec recommended)
 - e. Output Timeout < 10 msec (0 recommended)
 - f. Input Filter = Receive All
- 5.) Other CAN nodes may be connected to the same CAN bus, as long as they conform to the above properties.
- 6.) Use the CANX Restart button to re-establish connection with the NO_x sensor module if the CAN bus is disconnected and then connected again.
- 7.) NO_x bits must be converted to NO_x ppm at the RT level according to:
 - a. NO_x ppm = (bits * 0.05) – 200.
 - b. Data type of bits is I16.
 - c. The above conversion is performed by nox_rt_reva.vi.
- 8.) O₂ bits must be converted to O₂% at the RT level according to:
 - a. O₂% = (bits * 0.000514) – 12.
 - b. Data type of bits is U16.
 - c. The above conversion is performed by nox_rt_reva.vi.

Right click on the NI 9853 module within your project explorer and select **Properties**. A dialog will open and allow you to make the CAN module settings listed in item 4 above. The property dialogs are shown below in figure 5a and 5b.

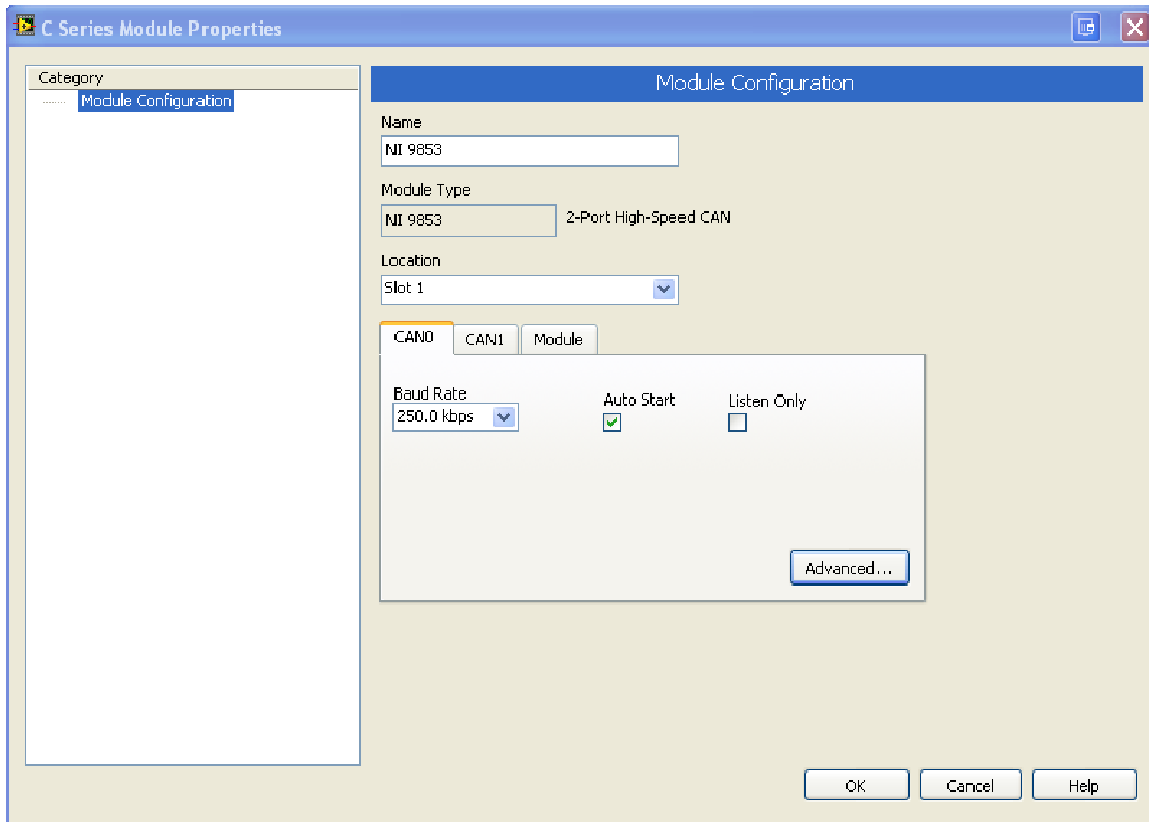


Figure 9 Top level property dialog for NI 9853 module

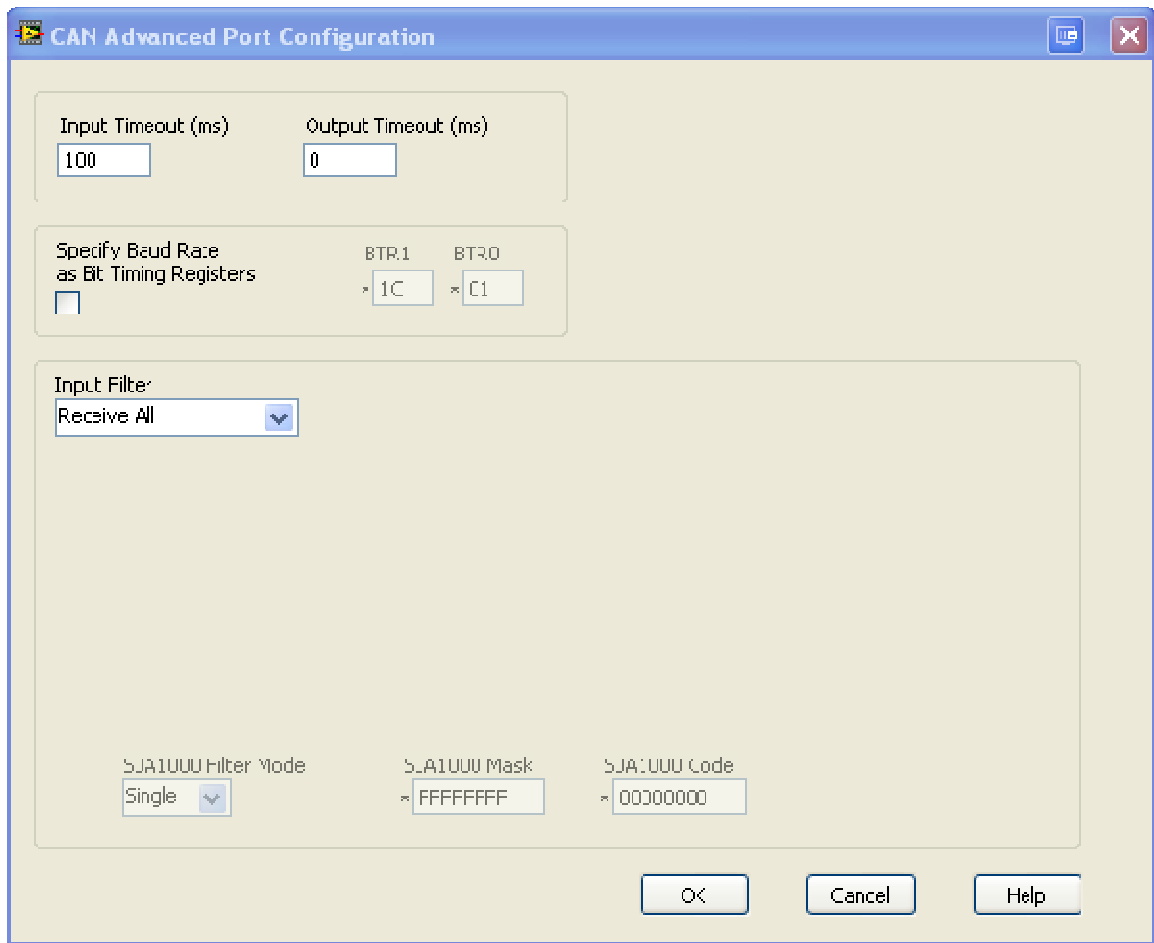


Figure 10 Advanced property dialog for NI 9853 module

If the CAN connection is expected to be disconnected at times while the NI 9853 CAN module is active, then it may be necessary to implement a method to reset the CAN channel being used. Do this by placing a FPGA I/O Method node which resets the CAN channel. The method node should be placed within a case statement and selected by a Boolean control. This Boolean control can be operated from the RT level programmatically.

Follow the NO_x_Example_RT.vi to properly interface to your FPGA application. Use the nox_rt_reva.vi to convert the data cluster from the nox_reva.vi to engineering units. It is not necessary to access the TxID or RxID. They should never change. In fact, they can be replaced by constants at the FPGA level. They are made available in case the default IDs change in the NO_x sensor module.

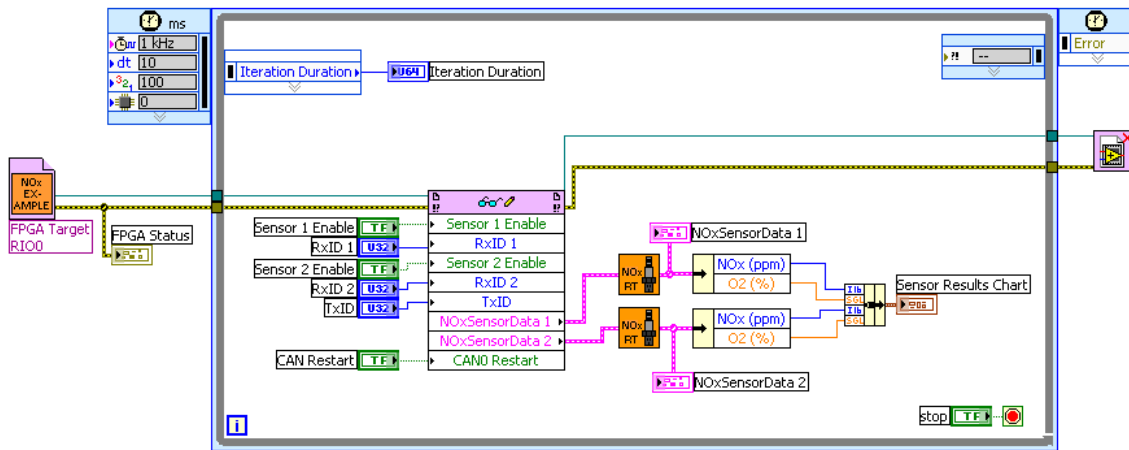


Figure 11 Example LabVIEW RT interface to LabVIEW FPGA NO_x_Example_FPGA.vi

cRIO Sub VI Documentation

nox_reva.vi

The NO_x sensor FPGA VI should be placed in its own while loop (as shown in the example). Do not place more than one NO_x sensor FPGA VI within the same while loop. Do not place the NO_x sensor FPGA VI within a single cycle loop.

The cRIO 9853 CAN Module properties should be set to the following:

Baud Rate = 250 Kbps
 Auto Start = Checked
 Listen Only = Unchecked
 Input Timeout > 50 msec (100 msec recommended)
 Output Timeout < 10 msec (0 recommended)
 Input Filter = Receive All

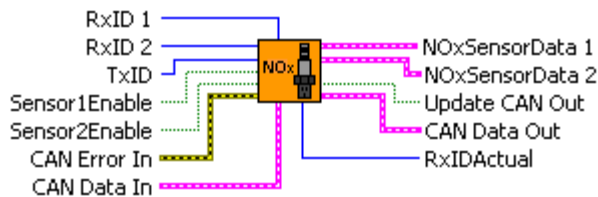
Other CAN nodes may be connected to the same CAN bus, as long as they conform to the above properties.

Use the FPGA CANX Restart method to re-establish connection with the NO_x sensor module if the CAN bus is disconnected and then connected again.

The following conversions are implemented at the RT level by the nox_rt_revx.vi.

NO_x ppm = (bits * 0.05) - 200.

O₂% = (bits * 0.000514) - 12.



U32 **TxID** This is the Transmit ID for the NI CAN module.

This is the Receive ID for the NO_x module.










By default the NO_x module receive ID is 18FEDF00h, corresponding to PGN 65247. This PGN contains the heater control command.

U32 **RxID 1** ID of the CAN message that should be received from the Sensor Control Module for channel 1.

By default the NO_x module transmit ID is 18F00F52h, corresponding to PGN 61455. This PGN contains the NO_x, O₂ and status information from the NO_x module.

U32 **RxID 2** ID of the CAN message that should be received from the Sensor Control Module for channel 2.

By default the NO_x module transmit ID is 18F00E51h, corresponding to PGN 61455. This PGN contains the NO_x, O₂ and status information from the NO_x module.

-  **Sensor 1Enable** When TRUE, the primary(channel 1) NO_x sensor module is requested to turn on the sensor heater and begin controlling the sensor to high temperature. When FALSE, the NO_x sensor module is requested to place the heater circuit in preheating mode and begin controlling to 80 C to 120 C.
-  **Sensor 2Enable** When TRUE, the secondary(channel 2) NO_x sensor module is requested to turn on the sensor heater and begin controlling the sensor to high temperature. When FALSE, the NO_x sensor module is requested to place the heater circuit in preheating mode and begin controlling to 80 C to 120 C.
-  **CAN Error In** Receives the error cluster wire from the error output terminal of the FPGA CAN I/O node (configured as "Read"). Right click on the FPGA CAN I/O Node and click on Show Error Terminals.
-  **CAN Data In** Receives the cluster wire from the FPGA CAN I/O node. The FPGA CAN I/O node should be configured as "Read".
-  **NO_xSensorData1** This cluster contains NO_x, O₂ and status data for sensor 1. This terminal should be terminated with a cluster within the FPGA block diagram and made available to the LabVIEW RT block diagram to be connected to the "NO_xSensorData From FPGA" input terminal of the nox_rt_revx.vi.
-  **NO_xSensorData2** This cluster contains NO_x, O₂ and status data for sensor 2. This terminal should be terminated with a cluster within the FPGA block diagram and made available to the LabVIEW RT block diagram to be connected to the "NO_xSensorData From FPGA" input terminal of the nox_rt_revx.vi.
-  **Update CAN Out** Must be wired to the selector terminal of a case statement structure. The TRUE case of the case statement must contain the FPGA CAN I/O node configured for "Write".
-  **CAN Data Out** Output cluster to be wired to the FPGA CAN I/O node. The FPGA CAN I/O node should be configured as "Write".
-  **RxIDActual** This is the actual received ID for the NI CAN module. This is useful to determine what message IDs are actually being received.

nox_rt_reva.vi

This VI is to be implemented at the RT level. It receives data from the FPGA VI "NOxSensorData" output cluster and properly scales and offsets the results.



NOxSensorDataFromFPGA This terminal should be wired from the "NOxSensorData" cluster output from the FPGA VI.

NOxSensorData Final processed result from the NOx Sensor.

NOx (ppm) Parts per million of oxides of nitrogen measured in the exhaust gas.

O2 (%) Percent oxygen content of the exhaust gas.

Sensor Supply Status Provides status information about the NOx sensor power supply.

The following status messages are provided:

- NOT IN RANGE
- SUPPLY IN RANGE
- ERROR
- NOT AVAILABLE

Sensor Temperature Status Provides status information about the NOx sensor heater temperature.

The following status messages are provided:

- NOT AT TEMPERATURE
- AT TEMPERATURE
- ERROR
- NOT AVAILABLE

NOx Signal Status Provides status information about the NOx sensor NOx ppm measurement.

The following status messages are provided:

- NOT VALID
- VALID
- ERROR
- NOT AVAILABLE

O2 Signal Status Provides status information about the NOx sensor O2 measurement.

The following status messages are provided:

- NOT VALID
- VALID
- ERROR
- NOT AVAILABLE

Heater Mode Status Provides status information about the NOx sensor heater control mode.

The following status messages are provided:

AUTO MODE
HEATUP SLOPE 3 OR 4
HEATUP SLOPE 1 OR 2
HEATER OFF / PREHEAT

U8 **Heater Circuit Status** Provides status information about the NO_x sensor heater circuit.

The following status messages are provided:
OPEN WIRE
SHORT CIRCUIT
NO ERROR.

U8 **NO_x Circuit Status** Provides status information about the NO_x sensor NO_x measurement circuit.

The following status messages are provided:
OPEN WIRE
SHORT CIRCUIT
NO ERROR.

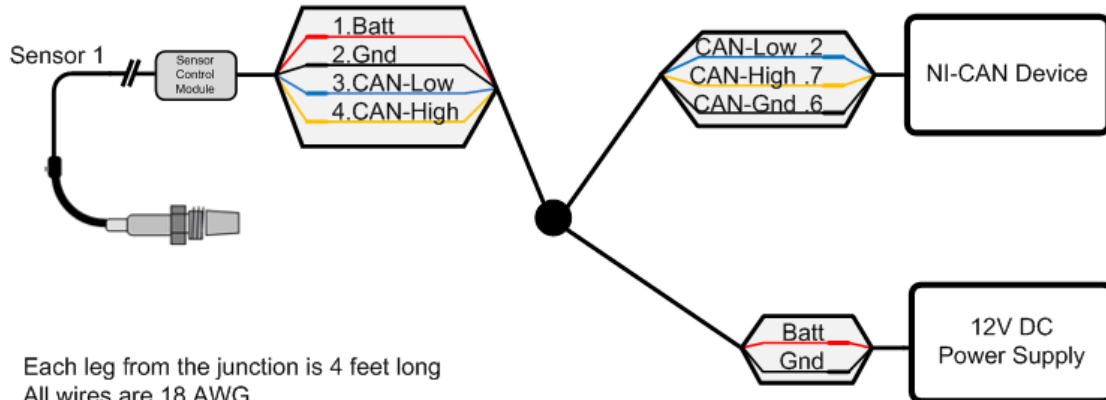
U8 **O2 Circuit Status** Provides status information about the NO_x sensor O2 measurement circuit.

The following status messages are provided:
OPEN WIRE
SHORT CIRCUIT
NO ERROR.

TF **NO_x Sensor Module Present** Indicates whether the NO_x sensor is powered, properly connected, and detected on the CAN bus.

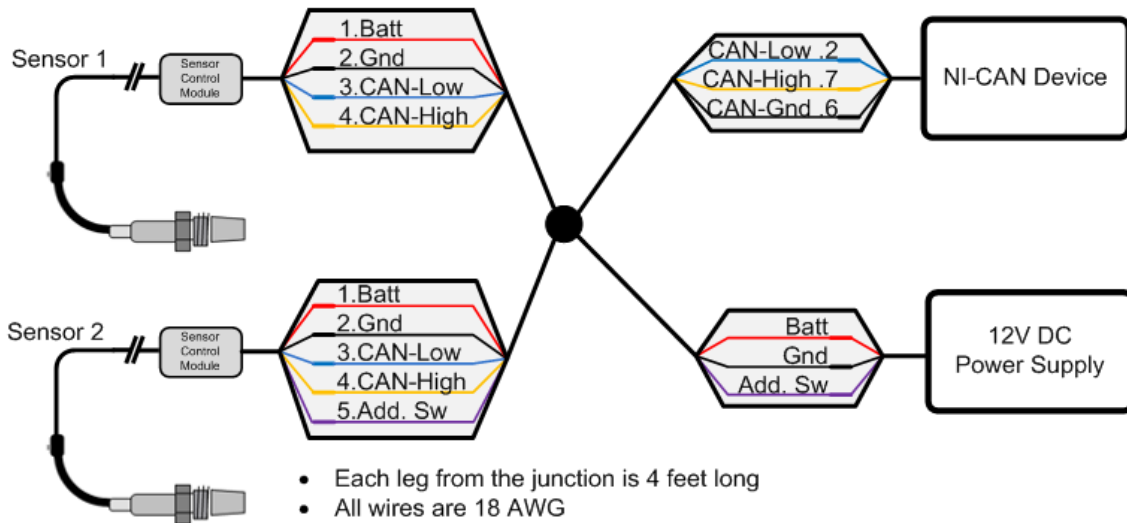
Wiring Harness Documentation

1-Sensor Harness



- Each leg from the junction is 4 feet long
- All wires are 18 AWG
- 120ohm resistors across CAN-Low and CAN-High near the sensor
- NI-CAN devices connect to the harness through a female DB-9 connector.
- CAN-Ground is connected to Ground near junction.

2-Sensor Harness



- Each leg from the junction is 4 feet long
- All wires are 18 AWG
- 120ohm resistors across CAN-Low and CAN-High near sensor 1 and 2
- NI-CAN devices connect to the harness through a female DB-9 connector.
- CAN-Ground is connected to Ground near junction.
- Add. Sw should be connected to ground at the power supply