Single-Board RIO Device Reference and Procedures

2024-04-29

Contents

Single-Board RIO Device Reference and Procedures

Use this book as a reference for information about using Single-Board RIO (sbRIO) devices with NI CompactRIO Device Drivers in LabVIEW.

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The **LabVIEW Help** highlights this topic in the **Contents** tab so you can navigate the related topics.

sbRIO-9603/9608/9609/9628/9638/9629 CAN API

FPGA I/O Node

You can use an FPGA I/O Node for either CAN Input or CAN Output.

Terminals in Software

You can select the following terminals for this device.

Terminal	Description
CAN0	CAN Port 0

I/O Methods

You can use the following I/O methods on the sbRIO-9603/9608/9609/9628/9638/9629.

Method	Description
Abort Transmit	Abort a pending CAN frame transmission.
Reset	Reset the CAN port to the same state as when th e FPGA VI started running.
Start	Start communication.

Stop	Stop communication.
Wait on Comm State Change	Wait for a change in the communication state of the CAN port (Comm State property).
Wait on Transceiver Wakeup	Wait for the Transceiver Mode property to change from Sleep to Normal mode due to a remote wakeup (bus activity) or local wakeup.
Wait on Transmit Complete	Wait for **all** frames written to CAN Output to complete transmission.

I/O Properties

Use the FPGA I/O Property Node to access the following properties with the sbRIO-9603/9608/9609/9628/9638/9629.

Property	Description	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
Baud Rate	Specifies the baud rate and default sample point for CAN bus.	No	No	Yes	No
Baud Rate Advanced	Specifies custom baud rate and sample point for CAN bus.	No	No	Yes	No
FD Baud Rate	Specifies the baud rate and default sample point for CAN bus with flexible data rate.	No	No	Yes	No
FD Baud Rate Advanced	Specifies custom baud rate and sample point for CAN bus with flexible data rate.	No	No	Yes	No
Comm State	Describes the current commu	Yes	Yes	N/A	N/A

	nication state of the CAN controller.				
Listen Only	Controls listen-only mode for passive monitoring/logging.	No	No	Yes	No
I/O Mode	Sets the CAN operating mode to CAN(default), CAN FD or CAN FD + BRS.	No	No	Yes	No
Input Timeout (ms)	Specifies the time to wait for the reading of CAN frame using CAN Input. The resolution is in milliseconds.	No	No	Yes	No
Output Timeout (ms)	Specifies the time to wait for the writing of CAN frame using CAN Output. The resolution is in milliseconds .	No	No	Yes	No
Log Bus Errors	Enables the logging of bus errors as frames that can be read using the CAN Input node.	No	No	Yes	Yes
Log Transceiver Faults	Enables the logging of transceiver faults as frames that can be read using the CAN Input node.	No	No	Yes	Yes

Self Reception	Specifies whether to echo successfully transmitted CAN frames to be read using CAN Input.	No	No	Yes	Yes
Single Shot Transmit	Specifies whether to retry failed CAN frame transmissions.	No	No	Yes	No
Receive Error Counter	Provides access to the CAN controller Receive Error Counter.	Yes	Yes	N/A	N/A
Transmit Error Counter	Provides access to the CAN controller Transmit Error Counter.	Yes	Yes	N/A	N/A
Transceiver Mode	Sets the mode for the CAN transceiver and the associated mode in the TCAN4550 CAN controller.	No	Yes	No	Yes

CAN FPGA I/O Properties Dialog Box

The CAN FPGA I/O Properties dialog box includes the following properties that you can use to configure the CAN functionality. To open the dialog box, right-click **CAN0** in the Project Explorer window and select **Properties**.

Baud Rate

This property specifies the Baud Rate for the transceiver when in classical CAN mode or when transmitting data before the BRS bit when in CAN FD + BRS mode. You must always set Baud Rate even when the I/O Mode is set to CAN FD + BRS because the CAN FD + BRS protocol always has a portion of the frame that transfer in classical CAN mode.

Typical baud rates are listed as kilobits per second. The supported values are 5.00, 6.150, 10.00, 12.5, 16.00, 20.00, 25.00, 31.25, 33.33, 40.00, 50.00, 62.50, 80.00, 83.33, 100.0, 125.0, 160.0, 200.0, 250.0, 400.0, 500.0, 800.0, and 1000. These bit fields are selected to create the desired baud rate with a default sample point of 87.5%. The default baud rate is 500.0 kb/s.

If you prefer a custom sample point, select <Custom> and the Custom Baud Rate dialog box will appear that you can use to determine the values to put in this field to obtain the desired sample point.

FD Baud Rate

This property specifies the baud rate for the transceiver when in FD + BRS CAN mode while transmitting data after the BRS bit. You need to set the FD Baud Rate or FD Baud Rate Advanced properties only when the I/O Mode is set to CAN FD + BRS.

Typical FD baud rates are listed as kilobits per second. The supported values are 200, 250, 400, 500, 800, 1000, 1250, 1600, 2000, 2500, 4000, and 5000. These bit fields are selected to create the desired baud rate with a default sample point of 87.5%. The default FD Baud Rate is 500.0 kb/s.

If you prefer a custom sample point, select <Custom> and the Custom FD Baud Rate dialog box will appear that you can use to determine the values to put in this field to obtain the desired sample point.

I/O Mode

This configures the CAN operating mode. The values are:

CAN(default)	Default CAN 2.0
CAN FD	CAN FD mode as specified in the CAN with Flexible Data-Rate specification, version 1.0
CAN FD + BRS	CAN FD as specified in the CAN with Flexible Data-Rate specification, version 1.0, with the optional Baud Rate Switching enabled.

Input Timeout (ms)

The Input Timeout (ms) property specifies how long to wait for a new frame to be received. It is a signed 32-bit integer with a resolution of milliseconds, thus allowing a maximum value of approximately 25 days. Special values of 0 (do not wait) and -1 (wait indefinitely) are supported. The default input timeout is 10000 ms.

If you specify an Input Timeout (ms) of 0, the CAN Input node will simply check to see if a new frame has arrived (non-blocking). If a new frame exists, CAN Input returns the frame with an error status of FALSE (success). If no new frame exists, CAN Input returns an error status of TRUE (error). Therefore, you must enable Error Terminals for the CAN Input node in order to poll for new frames when using an input timeout of 0. When an error is returned, you must invoke the CAN Input node again at a later time to ensure no data is lost.

Output Timeout (ms)

The Output Timeout (ms) property specifies how long to wait for a new element to become available, which occurs when a frame from a previous CAN Output transmits successfully onto the network. If you specify an Output Timeout (ms) of 0, the CAN Output node returns an error status of TRUE (error) if a new element is not available in the FIFO (non-blocking). When an error is returned, you must attempt CAN Output of the same frame again at a later time. Special values of 0 (do not wait) and -1 (wait indefinitely) are supported. The default output timeout is 10000 ms.

Auto Start

This Boolean property indicates whether to invoke the Start method automatically when the FPGA VI runs. The default is TRUE (enabled).

When Auto Start is TRUE (enabled), the Start method is invoked automatically when the FPGA VI runs, and the Stop method is invoked automatically when the FPGA VI stops running. This enables your FPGA VI diagram to begin using CAN Input and CAN Output nodes without first using an explicit Start method.

When Auto Start is FALSE (disabled), the Start method is not invoked automatically. You must use the Start method in your FPGA VI diagram in order to start communication.

Log Bus Errors

This Boolean control indicates whether to enable the Log Bus Errors feature for logging of bus errors as frames that can be read using the CAN Input node. The default is FALSE(disabled).

When Log Bus Errors is TRUE (enabled), the bus errors will be logged as frames and can be read using the CAN Input node.

When Log Bus Errors is FALSE (disabled), the bus errors will not be logged and cannot be read using the CAN Input node (default).

Log Transceiver Faults

This Boolean control indicates whether to enable the Log Transceiver Faults feature for logging of transceiver faults as frames that can be read using the CAN Input node. The default is FALSE(disabled).

When Log Transceiver Faults is TRUE (enabled), the logging is enabled and the transceiver faults will be logged as frames which can be monitored using the CAN Input node.

When Log Transceiver Faults is FALSE (disabled), the transceiver faults will not be logged as frames.

Self Reception

This Boolean control indicates whether to enable the Self Reception feature to echo successfully transmitted CAN frames to be read using CAN Input node. The default is FALSE(disabled).

When Self Reception is TRUE (enabled), the transmits are echoed.

When Self Reception is FALSE (disabled), the transmits are not echoed.

Single Shot Transmit

This Boolean control indicates whether to enable the Single Shot Transmit feature to prevent failed CAN frame transmissions from retrying. The default is FALSE (disabled).

When Single Shot Transmit is TRUE (enabled), single-shot is enabled. If a CAN frame is not transmitted successfully, the CAN controller will not retry.

When Single Shot Transmit is FALSE (disabled), the failed CAN frame transmissions are automatically retried.

Listen Only

This Boolean control indicates whether to enable the Listen Only feature for passive monitoring of the network. The default is FALSE (disabled).

When Listen Only is FALSE (disabled), you can transmit CAN messages normally using CAN Output. When CAN messages are received, those messages are acknowledged.

When Listen Only is TRUE (enabled), you cannot transmit CAN messages. When CAN messages are received, those messages are not acknowledged. The TCAN4550 CAN controller enters error passive state when Listen Only is enabled. Checking Listen Only enables passive monitoring of network traffic, which can be useful for debugging scenarios in which only one device exists on the network.

CAN Input

Wait for a CAN frame to be received, then return that frame.

Complete the following steps to read from a CAN port using CAN Input.

1. Use the Add Targets and Devices dialog box to add the CAN module to your CompactRIO configuration. Right-click the CAN module in the Project Explorer window and select Properties to specify the configuration properties for the module.
2. Create an FPGA I/O item for CAN by right-clicking the FPGA target under **My Computer** in the **Project Explorer** window, selecting **New»FPGA I/O**, and then adding **CAN0** to the FPGA I/O list. Refer to New FPGA I/O dialog box (FPGA Module) for more information about creating and adding FPGA I/O to your CompactRIO configuration.
3. Place an I/O Node from the FPGA I/O palette onto your FPGA VI diagram.
4. Right-click the I/O Node and select **Select FPGA I/O»CAN0**.

5. Right-click the I/O Node and select **Change to Read** to use the I/O node for CAN Input.
6. Right-click the I/O Node and choose **Properties** to select the Data Type.

Start communication on the CAN port prior to using CAN Input. Enable **Auto Start** in the Module Configuration, or invoke the Start method to start communication.

The **Input Timeout (ms)** of the CAN Advanced Port Configuration dialog box specifies how long to wait for a new frame to be received. If you specify **Input Timeout (ms)** of 0, the CAN Input node will simply check to see if a new frame has arrived (non-blocking).

CAN Input can store the result of two received packets. Receiving an additional packet without calling CAN Input will cause an overflow error. NI recommends placing CAN Input in a loop that is executed when the device receives a data packet.

For information on how CAN Input arbitrates with other CAN nodes, refer to Arbitration.

Note You can add more than one CAN input port to the FPGA I/O node, and also add analog and digital inputs to the same FPGA I/O node by right-clicking the FPGA I/O node and selecting **Add Element**. Right-click the created I/O name and choose **Select FPGA** I/O to specify the I/O name. If you use more than one input per node, LabVIEW will execute them sequentially starting with the top input. If one input is blocked while waiting for the input, subsequent inputs will be delayed. For example, if you read an Analog or Digital Input and CAN0 in the same input node, that node will wait for a frame to be received on CAN0 and wait for Analog or Digital data to be available. Accessing the CAN0 port using a distinct CAN Input Node is more efficient.

Properties Dialog

The **Data Type** selects the type for the CAN frame returned by the CAN Input node. The available values are **Cluster** (default) and **Array of U32**.

The benefits of using the **Cluster** data type are:

- Simpler block diagram (easy to use)
- Fast and efficient, if you limit use to the block diagram only (not front panel)

The benefit of the **Array of U32** data type, and handshaking integers to the host VI one at a time, is that it is faster for transfer to/from host VI.

Refer to the **Using Clusters and Arrays in LabVIEW FPGA** Application Note for more information on the use of clusters and arrays.

Node Inputs

	Error In Optional. Not shown by default. Right-click the node and select Error Terminals to enable.

Node Outputs

	Error Out Optional. Not shown by default. Right-click the node and select Error Terminals to enable. Unless you set the **Input Timeout (ms)** to –1 to wait indefinitely, you must use the error terminals to determine whether the CAN Input successfully received a CAN frame.
	CANx If the **Data Type** is defined as **Cluster** (default), the CAN frame is represented as a cluster with the following elements:

U32	Timestamp High

	U32	**Timestamp Low** Timestamp of when the frame was received. The timestamp is acquired at the end of the CAN frame. The high and low U32 represent a single U64 timestamp. The timestamp is large enough to avoid handling rollover. The timestamp is zero-based (relative). The time starts ticking at zero when CAN communication starts. The resolution is 100 ns.
	U32	**Identifier** Arbitration ID. If bit 29 (20000000 hexadecimal) is clear, this ID uses the standard format (11-bit). If bit 29 is set, this ID uses extended format (29-bit).
	U8	**Type** Type of frame: Data Frame 0 Remote Frame 1

Bus Error	6
Transceiver Fault	7
CAN FD Data Frame	16
CAN FD+BRS Data Frame	24

▸U8	**InfoA** Reserved for future use.
▸U8	**InfoB** Reserved for future use.
▸U8	**Data Length** For data frames, this indicates the number of bytes in **Data** (0–8 for CAN and 0–64 for CAN FD). For remote frames, this indicates the number of bytes requested.
[U8]	**Data** For data frames, this provides the data

bytes. The array uses a fixed size of 64 bytes. Values above Data Length are not valid and should be ignored. For remote frames, the values contained in **Data** are not valid and should be ignored.

If the **Data Type** is defined as **Array of U32**, the CAN frame is represented as the following Array of U32 values:

Most significant byte	...	...	Least significant byte
Timestamp (upper U32)			

Timestamp (lower U32)			
Identifier			
Type	Info A	Info B	Data Length
Data[0]	Data[1]	Data[2]	Data[3]
Data[4]	Data[5]	Data[6]	Data[7]
Data[8]	Data[9]	Data[10]	Data[11]
Data[12]	Data[13]	Data[14]	Data[15]
Data[16]	Data[17]	Data[18]	Data[19]
Data[20]	Data[21]	Data[22]	Data[23]

Data[24]	Data[25]	Data[26]	Data[27]
Data[28]	Data[29]	Data[30]	Data[31]
Data[32]	Data[33]	Data[34]	Data[35]
Data[36]	Data[37]	Data[38]	Data[39]
Data[40]	Data[41]	Data[42]	Data[43]
Data[44]	Data[45]	Data[46]	Data[47]
Data[48]	Data[49]	Data[50]	Data[51]
Data[52]	Data[53]	Data[54]	Data[55]

Data[56]	Data[57]	Data[58]	Data[59]	The meaning of each element is the same as the **Cluster** data type.
Data[60]	Data[61]	Data[62]	Data[63]	

CAN Output

Write a CAN frame to be transmitted.

Complete the following steps to write to a CAN port using CAN Output.

1. Use the Add Targets and Devices dialog box to add the CAN module to your CompactRIO configuration. Right-click the CAN module in the Project Explorer window and select Properties to specify the configuration properties for the module.
2. Create an FPGA I/O item for CAN by right-clicking the FPGA target under **My Computer** in the **Project Explorer** window, selecting **New»FPGA I/O**, and then adding **CAN0** to the FPGA I/O list. Refer to New FPGA I/O dialog box (FPGA Module) for more information about creating and adding FPGA I/O to your CompactRIO configuration.
3. Place an I/O Node from the FPGA I/O palette onto your FPGA VI diagram.
4. Right-click the I/O Node and select **Select FPGA I/O»CAN0**.
5. Right-click the I/O Node and select **Change to Write** to use the I/O node for CAN Output.

6. Right-click the I/O Node and choose **Properties** to select the Data Type.

Start communication on the CAN port prior to using CAN Output. Enable **Auto Start** in the Module Configuration, or invoking the Start method to start communication.

The communication path from LabVIEW FPGA to the CAN port on the module is implemented as a FIFO. The CAN Output node waits for an available element in the FIFO, then writes the frame to the FIFO. If no other CAN frame exists in the FIFO, the CAN frame will begin to transmit immediately.

CAN Output does not wait for the requested transmit to complete on the network (acknowledgment). Call CAN Output followed by Wait on Transmit Complete (I/O Method) if you want to transmit CAN frames one at a time.

Write multiple frames to the output FIFO if you want to transmit CAN frames as fast as possible. The CAN module will submit these frames to the TCAN4550 controller as fast as possible.

When the output FIFO is full, the CAN Output node waits for an element to become available, then writes the frame to the FIFO. The **Output Timeout (ms)** of the CAN Advanced Port Configuration dialog box specifies how long to wait for a new element to become available, which occurs when a frame from a previous CAN Output transmits successfully onto the network. If you specify **Output Timeout (ms)** of 0, the CAN Output node returns an error status of TRUE (error) if a new element is not available (non-blocking). When an error is returned, you must attempt CAN Output of the same frame again at a later time.

In addition to the CAN Output node, the **Output Timeout (ms)** also applies to most methods and properties. For more information on the output path from LabVIEW FPGA to the CAN port, refer to Arbitration.

Note You can add more than one CAN output port to the FPGA I/O node, and also add analog and digital outputs to the same FPGA I/O node by right-clicking the FPGA I/O node and selecting **Add Element**. Right-click the created I/O name and select **Select FPGA I/O** to specify the I/O name. If you use more than one output per node, LabVIEW will execute them sequentially starting with the top output. If one output is blocked while waiting for the output, subsequent outputs will be delayed.

Note The Transmit Packet Type may not exceed the mode set by the I/O Mode property.

Properties Dialog

Selects the **Data Type** for the CAN frame written to the CAN Input node. The available values are **Cluster** (default) and **Array of U32**.

The benefits of using the **Cluster** data type are:

- Simpler block diagram (easy to use)
- Fast and efficient, if you limit use to the block diagram only (not front panel)

The benefits of the **Array of U32** data type, and handshaking integers to the host VI one at a time, are:

- Faster for transfer to/from host VI
- Fewer FPGA gates

Refer to the **Using Clusters and Arrays in LabVIEW FPGA** Application Note for more information on the use of clusters and arrays.

Node Inputs

	Error In Optional. Not shown by default. Right-click the node and select Error Terminals to enable.
	CANx If the **Data Type** is defined as **Cluster** (default), the CAN frame is represented as a cluster with the following elements:
	U32 **Timestamp High** Ignored. The transmit request always occurs

	as soon as possible (not timed).
U32	**Timestamp Low** Ignored. The transmit request always occurs as soon as possible (not timed).
U32	**Identifier** Arbitration ID. If bit 29 (20000000 hexadecimal) is clear, this ID uses the standard format (11-bit). If bit 29 is set, this ID uses extended format (29-bit).
U8	**Type** Type of frame: **Data Frame** 0 **Remote Frame** 1 **CAN FD Data Frame** 16 **CAN FD+BRS Data Frame** 24

	U8	**InfoA** Reserved for future use.
	U8	**InfoB** Reserved for future use.
	U8	**Data Length** For data frames, this indicates the number of bytes in **Data** (0–8 for CAN and 0–64 for CAN FD.). For remote frames, this indicates the number of bytes requested.
	[U8]	**Data** For data frames, this provides the data bytes. The array uses a fixed size of 64 bytes.

If the **Data Type** is defined as **Array of U32** , the CAN frame is represented as the following Array of U32 values:

M os	...	...	L e

t significant byte			ast significant byte
Timestamp (upper U32)			
Timestamp (lower U32)			
Identifier			
Type	Info A	Info B	Data Length
Data[0]	Data[1]	Data[2]	Data[3]
Data[4]	Data[5]	Data[6]	Data[7]
Data[8]	Data[9]	Data[10]	Data[11]

Data[12]	Data[13]	Data[14]	Data[15]
Data[16]	Data[17]	Data[18]	Data[19]
Data[20]	Data[21]	Data[22]	Data[23]
Data[24]	Data[25]	Data[26]	Data[27]
Data[28]	Data[29]	Data[30]	Data[31]
Data[32]	Data[33]	Data[34]	Data[35]
Data[36]	Data[37]	Data[38]	Data[39]
Data[40]	Data[41]	Data[42]	Data[43]

Data[44]	Data[45]	Data[46]	Data[47]
Data[48]	Data[49]	Data[50]	Data[51]
Data[52]	Data[53]	Data[54]	Data[55]
Data[56]	Data[57]	Data[58]	Data[59]
Data[60]	Data[61]	Data[62]	Data[63]

The meaning of each element is the same as the **Cluster** data type.

Node Outputs

Error Out

Optional. Not shown by default. Right-click the node and select Error Terminals to enable.

	Unless you set the **Output Timeout (ms)** to -1 to wait indefinitely, you must use the error terminals to determine whether the CAN Output successfully submitted a CAN frame for transmit.

Custom Baud Rate Dialog Box

This dialog configures advanced properties for the CAN port when in classical CAN or FD + BRS CAN mode. You can access this dialog from the FPGA I/O Properties dialog box.

Baud Rate

Specifies the custom Baud Rate for the transceiver when in classical CAN mode or when transmitting data before the BRS bit in CAN FD + BRS mode. The default Baud Rate is 500 kb/s.

Bit Timing Register

Specifies the custom CAN baud rate by setting the nominal Bit Timing and Prescaler register of the CAN controller. The fields are:

- **Time quantum (Tq)**—Programs the baud rate prescaler. Valid values are 25 to 12800, in increments of 0x19 (25 decimal). The default Tq is 50.
- **Synchronization Jump Width (SJW)**—Valid values are 0 to 127. The default SJW is 4. The actual hardware interpretation of this value is one more than the programmed value.
- **Time Segment 1 (TSEG1)**—Time segment before the sample point. Valid values are 1 to 0xFF (1 to 255 decimal). The default TSEG1 is 33. The actual hardware interpretation of this value is one more than the programmed value.
- **Time Segment 2 (TSEG2)**—Time segment after the sample point. Valid values are 0 to 0x7F (0 to 127 decimal). The default TSEG2 is 4. The actual hardware interpretation of this value is one more than the programmed value.

Sample Point

Specifies the custom sample point in classical CAN mode or when transmitting data before the BRS bit in CAN FD + BRS mode. The default sample point is 87.5%.

Custom FD Baud Rate Dialog Box

This dialog box allows you to configure advanced properties for the CAN port when in FD or FD + BRS CAN mode. You can access this dialog from the FPGA I/O Properties Page.

FD Baud Rate

Specifies the custom Baud Rate for the transceiver when transmitting data after the BRS bit in FD + BRS CAN mode. The default FD Baud Rate is 500 kb/s.

Bit Timing Register

Specifies the custom CAN FD baud rate with flexible data rate by setting the data Bit Timing and Prescaler register of the CAN controller. The fields are:

- **Time quantum (Tq)**—Programs the baud rate prescaler. Valid values are 25 to 800, in increments of 25 ns. The default Tq is 100.
- **Synchronization Jump Width (SJW)**—Valid values are 0 to 15. The default SJW is 3. The actual hardware interpretation of this value is one more than the programmed value.
- **Time Segment 1 (TSEG1)**—Time segment before the sample point. Valid values are 0 to 31. The default TSEG1 is 14. The actual hardware interpretation of this value is one more than the programmed value.
- **Time Segment 2 (TSEG2)**—Time segment after the sample point. Valid values are 0 to 15. The default TSEG2 is 3. The actual hardware interpretation of this value is one more than the programmed value.
- **Transmitter Delay Compensation (TDC)**—Enables or disables this feature.

Unchecked (default)	TDC disabled
Checked	TDC enabled

- **Transmitter Delay Compensation Offset (TDCO)**—Defines the distance between the delay from transmit to receive point and secondary sample point. Valid values are 0 to 127. The default TDCO is 0.
- **Transmitter Delay Compensation Filter Window Length (TDCF)**—Defines the minimum value for the secondary sample point position. It is enabled when TDCF is greater than TDCO. Valid values are 0 to 127. The default TDCF is 0.

Sample Point

Specifies the custom sample point in classical CAN FD mode or when transmitting data before the BRS bit in CAN FD + BRS mode. The default sample point is 80.0%.

I/O Properties

Complete the following steps to use an FPGA I/O Property node for CAN:

1. Use the Add Targets and Devices dialog box to add the CAN module to your CompactRIO configuration. Right-click the CAN module in the Project Explorer window and select Properties to specify the configuration properties for the module.
2. Create an FPGA I/O item for CAN by right-clicking the FPGA target under **My Computer** in the **Project Explorer** window, selecting **New»FPGA I/O»CAN0** to the FPGA I/O list. Refer to New FPGA I/O dialog box (FPGA Module) for more information about creating and adding FPGA I/O to your CompactRIO configuration.
3. Place an I/O Property Node from the FPGA I/O palette onto your FPGA VI diagram.
4. Right-click the I/O Property Node and select **Select Item»CAN0**.
5. Right-click the I/O Property Node and choose **Select Property** to specify the appropriate property from the list of available CAN properties.

For information on how the I/O Property Node arbitrates with other CAN nodes, refer to Arbitration.

The description of each property includes the following table.

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
<value>	<value>	<value>	<value>	<value>	<value>

Data type: Shows the icon for the LabVIEW data type used for this property.

Permissions: Read, Write, or Read/Write. If the property is Read/Write, you can right-click the property in the node and select **Change To Read** or **Change To Write** to select the direction. Otherwise, only one direction is supported.

Read While Stopped?: Yes or No to indicate whether you can read the property while CAN communication is stopped.

Read While Running?: Yes or No to indicate whether you can read the property while CAN communication is running.

Write While Stopped?: Yes or No to indicate whether you can write the property while CAN communication is stopped.

Write While Running?: Yes or No to indicate whether you can write the property while CAN communication is running.

Baud Rate

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U16	Write	No	No	Yes	No

Each CAN baud rate selection configures the correspond Baud Rate for the transceiver when in classical CAN mode or when transmitting data before the BRS bit when in CAN FD + BRS mode. This property must always be set even when I/O Mode is set to CAN FD + BRS as the CAN FD + BRS protocol always has a portion of the frame that transfer in classical CAN mode.

The supported values in kb/s are 5.00, 6.150, 10.00, 12.5, 16.00, 20.00, 25.00, 31.25, 33.33, 40.00, 50.00, 62.50, 80.00, 83.33, 100.0, 125.0, 160.0, 200.0, 250.0, 400.0, 500.0, 800.0, and 1000. These bit fields are selected to create the desired baud rate with a default sample point of 87.5 percent.

This property converts the baud rate as selected into corresponds Synchronization Jump Width (SJW), Nominal Bit Rate Prescaler (BRP), Nominal Time Segment Before Sample Point (TSEG1), and Nominal Time Segment After Sample Point (TSEG2).

Baud Rate Advanced

Data type		Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
Tq	U16	Write	No	No	Yes	No
SJW	U8					
TSEG1						
TSEG2						

This property allows you to configure the custom CAN baud rate by setting the nominal bit timing and prescaler register of the CAN controller. You can set the following fields.

Field	Description	Valid Values
Time quantum (Tq)	Programs the baud rate prescaler	25 to 12,800 in increments of 0x19 (25 decimal) Note that the actual hardware interpretation of this value is one more than the programmed value
Synchronization Jump Width (SJW)	The maximum number of time quanta by which a bit sampling period can be extended or shortened as a result of re-synchronization.	0 to 127
Time Segment 1 (TSEG1)	The time segment before the sample point.	1 to 0xFF (1 to 255 decimal)
Time Segment 2 (TSEG2)	The time segment after the sample point.	0 to 0x7F (0 to 127 decimal)

The time quantum is calculated by the following formula.

Time quantum = (**BRP** + 1) * 25

where **BRP** = nominal bit rate prescaler value

Use the custom sample point calculator on the FPGA I/O Properties page under **CAN0»Baud Rate»<Custom>** to determine the values to enter in this field to obtain the desired sample point.

Invalid Parameter or Error 65545 will be returned if you insert an invalid input.

FD Baud Rate

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U16	Write	No	No	Yes	No

This property sets the baud rate for the transceiver when in FD + BRS CAN mode when transmitting data after the BRS bit. FD Baud Rate or FD Baud Rate Advanced property only need to be set when I/O mode is set to CAN FD + BRS.

The supported values in kb/s are 200, 250, 400, 500, 800, 1000, 1250, 1600, 2000, 2500, 4000 and 5000. These bit fields are selected to create the desired baud rate with a default sample point of 87.5%

This property converts the baud rate as selected into corresponds Time Quantum (TQ), Synchronization Jump Width (SJW), Data Time Segment Before Sample Point (TSEG1), Data Time Segment After Sample Point (TSEG2), Transmitter Delay Compensation Enabler(TDC) and Transmitter Delay Compensation Filter Window Length (TDCF).

FD Baud Rate Advanced

Data type		Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
Tq	U16	Write	No	No	Yes	No
SJW	U8					
TSEG1						
TSEG2						
TDC	TF					
TDCF	U8					
TDCO						

This property allows you to configure the custom CAN FD baud rate by setting the nominal bit timing and prescaler register of the CAN controller. You can set the following fields.

Field	Description	Valid Values

Transmitter Delay Compensation (TDC)	Enables or disables this feature.	FALSE = disabled
		TRUE = enabled

Transmitter Delay Compensation Offset (TDCO)	Defines the distance between the delay from transmit to receive point and secondary sample point.	0 to 127
Transmitter Delay Compensation Filter Window Length (TDCF)	Defines the minimum value for the secondary sample point position. Enabled when TDCF is greater than TDCO.	0 to 127
Time quantum (Tq)	Programs the baud rate prescaler	25 to 800 in increments of 25 ns
Time Segment 1 (TSEG1)	The time segment before the sample point.	0 to 31
Time Segment 2 (TSEG2)	The time segment after the sample point.	0 to 15
Synchronization Jump Width (SJW)	The maximum number of time quanta by which a bit sampling period can be extended or shortened as a result of re-synchronization.	0 to 15 Note that the actual hardware interpretation of this value is one more than the programmed value

The time quantum is calculated by the following formula.

Time quantum = (**BRP** + 1) * 25

where	**BRP** = nominal bit rate prescaler value

Use the custom sample point calculator on the FPGA I/O Properties page under **CAN0»FD Baud Rate»<Custom>** to determine the values to enter in this field to obtain the desired sample point.

Invalid Parameter or Error 65545 will be returned if you insert an invalid input.

Comm State

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U8	Read	Yes	Yes	N/A	N/A

This property describes the current communication state of the CAN controller.

The values are:

0	Error Active
1	Error Passive
2	Bus Off

Use the Wait on Comm State Change method to detect changes in this property.

When you initially start communication, the CAN port will begin in the Error Active state as specified in the CAN standard (except when **Listen Only** is enabled). After bus errors cause a transition to Bus Off, that state will be retained until you call the Stop or Reset node. The Transmit Error Counter and Receive Error Counter will also reset to zero. If you are using the Stop node or Reset node with Auto Start disabled, you will need to call the Start node to continue communication in the Error Active state.

Listen Only

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
TF	Write	No	No	Yes	No

This property controls **Listen Only** mode for passive monitoring/logging.

FALSE disables **Listen Only** mode. Received frames are acknowledged, and frames can be transmitted using **CAN Output**.

TRUE enables **Listen Only** mode. The CAN port can only receive frames. The port does not transmit on the network. No acknowledgement is transmitted when a frame is received. When listen-only is enabled, the Comm State property begins in the Error Passive state.

I/O Mode

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U8	Write	No	No	Yes	No

This property configures the CAN operating mode. The values are:

CAN(default)

The default CAN 2.0 A/B standard I/O mode as defined in ISO 11898-1:2003. A fixed baud rate is used for transfer, and the payload length is limited to 8 bytes.

CAN FD

The CAN FD mode as specified in the CAN with Flexible Data-Rate specification, version 1.0. Payload lengths up to 64 are allowed, but they are transmitted at a single fixed baud rate.

CAN FD + BRS

The CAN FD as specified in the CAN with Flexible Data-Rate specification, version 1.0, with the optional Baud Rate Switching enabled. The same payload lengths as CAN FD mode are allowed; additionally, the data portion of the CAN frame is transferred at a different (higher) baud rate.

Input Timeout

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U32	Write	No	No	Yes	No

This property specifies the time to wait for the reading of CAN frame using CAN Input. The resolution is in milliseconds.

The timeout value of -1 (infinite wait) is supported.

Output Timeout (ms)

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U32	Write	No	No	Yes	No

This property specifies the time to wait for the writing of CAN frame using CAN Output. The resolution is in milliseconds.

Since this property set the wait for the pending transmission, use a timeout greater than zero. Using a timeout of 10,000 (ten seconds) is sufficient for most CAN baud rates (longer than a frame time).

The timeout value of -1 (infinite wait) is not supported, but you can use the largest positive I32 value, which provides a maximum of approximately 25 days.

Log Bus Errors

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
TF	Write	No	No	Yes	Yes

This property enables the logging of bus errors as frames that can be read using the CAN Input node.

The bus error frame is logged when the Texas Instruments TCAN4550 CAN controller detects a bus error.

FALSE indicates that the bus errors will not be logged and cannot be read using the CAN Input node (default). TRUE indicates that the bus errors will be logged as frames and can be read using the CAN Input node.

The bus error frame has the following format:

Arbitration ID	0
Data Length	5
Type	6
Data	**Bytes** 0—Comm State (see below) 1—Transmit Error Counter 2—Receive Error Counter 3—Detected Bus Error 4—X 5—X

	6—X 7—X

Note **X** means **Reserved** or **Don't Care**.

The first data byte (Comm State) indicates the current communication state of the CAN controller. The states are:

0—Error Active

1—Error Passive

2—Bus Off

The fourth data byte (Detected Bus Error) indicates the error detected during a frame transmission. The states are:

0—None

1—Stuff

2—Form

3—Ack

4—Bit 1

5—Bit 0

6—CRC

Log Transceiver Faults

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
TF	Write	No	No	Yes	Yes

This property enables the logging of transceiver faults as frames that can be read using the CAN Input node.

FALSE indicates that the transceiver faults will not be logged as frames (default). TRUE indicates that logging is enabled and the transceiver faults will be logged as frames which can be monitored using the CAN Input node.

The transceiver fault frame has the following format:

Arbitration ID	0
Data Length	1
Type	7
Data	**Bytes** 0—Transceiver fault (0=fault cleared, 1=fault present) 1—X 2—X 3—X 4—X 5—X 6—X 7—X

Note **X** means **Reserved** or **Don't Care**.

Self Reception

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
TF	Write	No	No	Yes	Yes

This property specifies whether to echo successfully transmitted CAN frames to be read using **CAN Input**. Each reception occurs just as if the frame were received from another CAN device. For self reception to operate properly, another CAN node must receive and acknowledge each transmit. If a transmitted frame is not successfully acknowledged, it is not echoed for input.

FALSE indicates transmits are not echoed (default), and TRUE indicates transmits are echoed.

Single Shot Transmit

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
TF	Write	No	No	Yes	Yes

This property specifies whether to retry failed CAN frame transmissions.

FALSE indicates standard CAN behavior, where failed CAN frame transmissions are automatically retried (default), and TRUE indicates single-shot. With single-shot enabled, if a CAN frame is not transmitted successfully, the CAN controller will not retry.

Receive Error Counter

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U8	Read	Yes	Yes	N/A	N/A

This property provides access to the CAN controller Receive Error Counter defined by the CAN standard.

Transmit Error Counter

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U8	Read	Yes	Yes	N/A	N/A

This property provides access to the CAN controller Transmit Error Counter defined by the CAN standard.

Transceiver Mode

Data type	Permissions	Read While Stopped?	Read While Running?	Write While Stopped?	Write While Running?
U8	Read/Write	No	Yes	No	Yes

This property sets the mode for the CAN transceiver, as well as the associated mode in the TCAN4550 CAN controller.

The values are:

0	Normal
1	Sleep

The Sleep mode places the TCAN4550 and the CAN transceiver into a low-power state. This low-power state has no effect on LabVIEW FPGA hardware. The Sleep mode can be set only when the CAN controller is running (not when stopped).

A **local wakeup** occurs when you set Transceiver Mode from Sleep back to Normal.

A **remote wakeup** occurs when a remote node transmits a CAN frame (referred to as the wakeup frame). The wakeup frame wakes up the transceiver and CAN controller chip. The wakeup frame is not received or acknowledged by the TCAN4550 CAN controller. When the wakeup frame ends, the CAN port enters Normal mode, and again receives and transmits CAN frames. If the node that transmitted the wakeup frame did not detect an acknowledgment (such as if other nodes were also waking), it will retry the transmission, and the retry will be received by the CAN port.

Once the transceiver is set into Sleep mode, you can invoke the Wait On Transceiver Wakeup method to determine when a remote wakeup occurs.

Reset

Reset the CAN port to the same state as when the FPGA VI started running.

This method resets the TCAN4550, and then restores all properties to their default values. If you enable **Auto Start** in the tool, this method will also re-start communication.

When a CAN controller enters the Bus Off state (see Comm State property), it no longer communicates. By performing a complete reset of the TCAN4550, this method recovers from the Bus Off state.

This method also clears the output FIFO as well as any pending CAN node (including all Wait methods). The reset affects only the specified CAN port, and has no effect on the other CAN port of the module.

Node Inputs

	Error In

	Optional. Not shown by default. To enable, right-click the node and select Error Terminals.

Node Outputs

	Error Out Optional. Not shown by default. To enable, right-click the node and select Error Terminals.

Start

Start communication.

Cases for using this method, as opposed to enabling **Auto Start** in Module Configuration:

- Change configuration properties within the diagram (such as the **Listen Only** property)
- Auto-baud algorithms (must Stop/Start to set the baud rate for each attempt)
- Start CAN communication on a trigger (analog or digital input)

Node Inputs

	Error In Optional. Not shown by default. To enable, right-click the node and select Error Terminals.

Node Outputs

	Error Out Optional. Not shown by default. To enable, right-click the node and select Error Terminals.

Stop

Stop communication.

This method also clears the output FIFO as well as any pending CAN node (including all Wait methods).

Node Inputs

	Error In Optional. Not shown by default. To enable, right-click the node and select Error Terminals.

Node Outputs

	Error Out Optional. Not shown by default. To enable, right-click the node and select Error Terminals.

Abort Transmit

Abort a pending CAN frame transmission.

Since the determinism of CAN depends on the prioritization of frames based on ID, the TCAN4550 CAN controller allows you to abort a pending transmit to replace it with a higher priority ID.

When implementing protocols that require abort, you must ensure that only one transmit is pending. Call **Wait on Transmit Complete** before (or after) each call to **CAN Output**. If more than one transmit is pending at the time **Abort Transmit** is called, the **Transmit Success** Boolean could refer to any one of those pending frames.

Node Inputs

	Error In Optional. Not shown by default. Right-click the node and select Error Terminals to enable.

Node Outputs

TF	**Transmit Success?**

	Indicates whether the pending transmit was successfully aborted, or transmitted on the network before it could be aborted. If the pending transmit aborted successfully, this returns FALSE. If the pending frame transmitted successfully on the network prior to the **Abort Transmit**, this returns TRUE.
	Error Out Optional. Not shown by default. Right-click the node and select Error Terminals to enable.

Wait on Comm State Change

Wait for a change in the communication state of the CAN port (Comm State property). This enables you to handle CAN communication state changes in a separate loop from your main CAN Input/Output loop.

If the **Comm State** property changes before you call Wait (relative to the previous Wait), the current Wait returns immediately. The **Comm State** is not considered to be changed when you first Start, and it does not change after Stop.

This method returns an error if invoked while communication is stopped.

Node Inputs

	Error In Optional. Not shown by default. To enable, right-click the node and select Error Terminals.
I32	**Timeout** Specifies the time to wait for the communication state to change. The resolution is in milliseconds. The special **Timeout** of 0 is used to poll for the changed state. If **Comm State** has changed

	from the previous call to Wait, the **Timed Out?** Boolean is FALSE. If **Comm State** has not changed, the **Timed Out?** Boolean is TRUE. The **Timeout** value of –1 (infinite wait) is supported.

Node Outputs

	Error Out Optional. Not shown by default. To enable, right-click the node and select Error Terminals.
TF	**Timed Out?** Indicates a timeout. When **Timed Out?** is TRUE, the error cluster indicates success (not a timeout error).
	Comm State Returns the new value of the **Comm State** property.

Wait on Transceiver Wakeup

Wait for the Transceiver Mode property to change from Sleep to Normal mode due to a remote wakeup (bus activity) or local wakeup. A local wakeup occurs if you use the I/O Property node to set **Transceiver Mode** property back to Normal.

If a wakeup occurs before you call Wait (relative to the previous Wait), the current Wait returns immediately. The wakeup is not considered to have occurred when you first Start, and it does not occur after Stop.

This method returns an error if invoked while communication is stopped.

Node Inputs

	Error In

	Optional. Not shown by default. To enable, right-click the node and select Error Terminals.
I32	**Timeout** Specifies the time to wait for the **Transceiver Mode** to change to Normal. The resolution is in milliseconds. The special **Timeout** of 0 is used to poll for the wakeup. If a wakeup has occurred since the previous call to Wait, the **Timed Out?** Boolean is FALSE. If **Transceiver Mode** is still Sleep, the **Timed Out?** Boolean is TRUE. The **Timeout** value of –1 (infinite wait) is supported.

Node Outputs

	Error Out Optional. Not shown by default. To enable, right-click the node and select Error Terminals.
TF	**Timed Out?** Indicates a timeout. When **Timed Out?** is TRUE, the error cluster indicates success (not a timeout error).

Wait on Transmit Complete

Wait for **all** frames written to CAN Output to complete transmission. You typically use this to determine when all frames have been acknowledged.

If all transmits are complete before you call Wait (relative to the previous Wait), the current wait returns immediately. All transmits are considered complete after you first Start. This method returns an error if invoked while communication is stopped.

Node Inputs

	Error In Optional. Not shown by default. To enable, right-click the node and select Error Terminals.
I32	**Timeout** Specifies the time to wait for all frames to complete transmission. The resolution is in milliseconds. The special **Timeout** of 0 is used to poll for transmit complete status. If all transmits are complete, the **Timed Out?** Boolean is FALSE. If one or more transmissions are pending (not successful or aborted), the **Timed Out?** Boolean is TRUE. The **Timeout** value of –1 (infinite wait) is supported.

Node Outputs

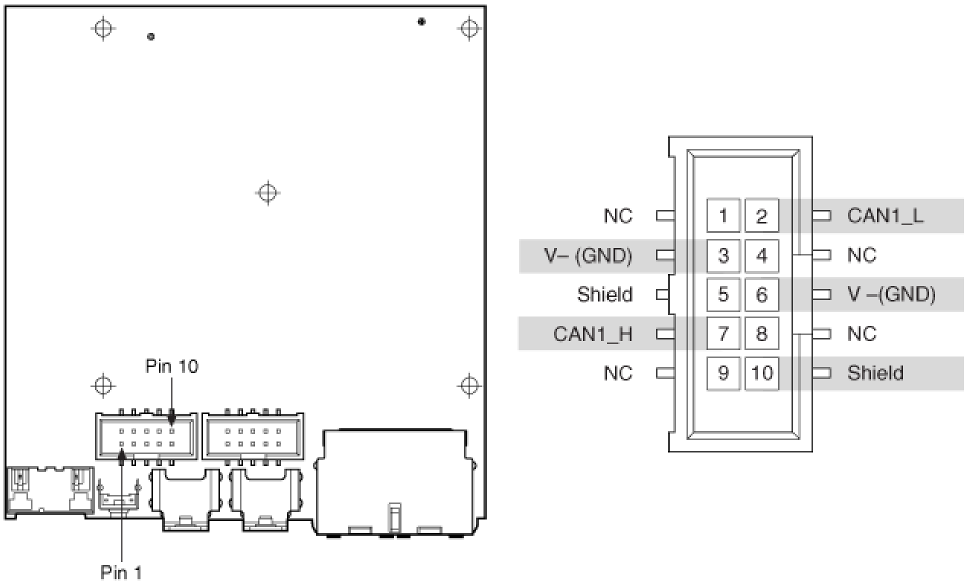
	Error Out Optional. Not shown by default. To enable, right-click the node and select Error Terminals.
TF	**Timed Out?** Indicates a timeout. When **Timed Out?** is TRUE, the error cluster indicates success (not a timeout error).

sbRIO-9603

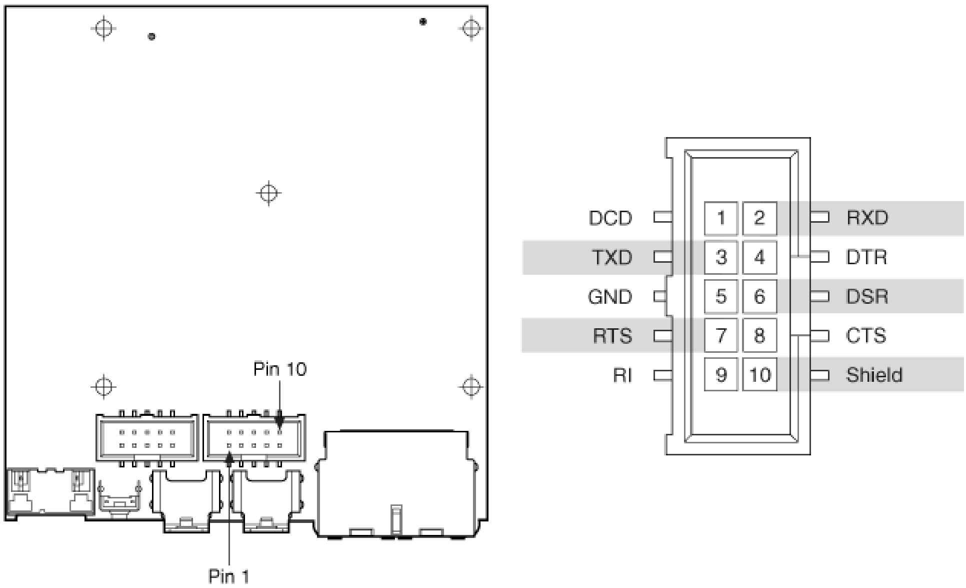
Single-Board Reconfigurable I/O (DIO), 1 RS-232 Serial port, 1 CAN port, Artix-7 75T FPGA

sbRIO-9603 Pinouts

Connector J3, CAN Port



Connector J4, Serial Port



Refer to the CompactRIO Related Documentation for further information about CompactRIO documentation.

Related Topics

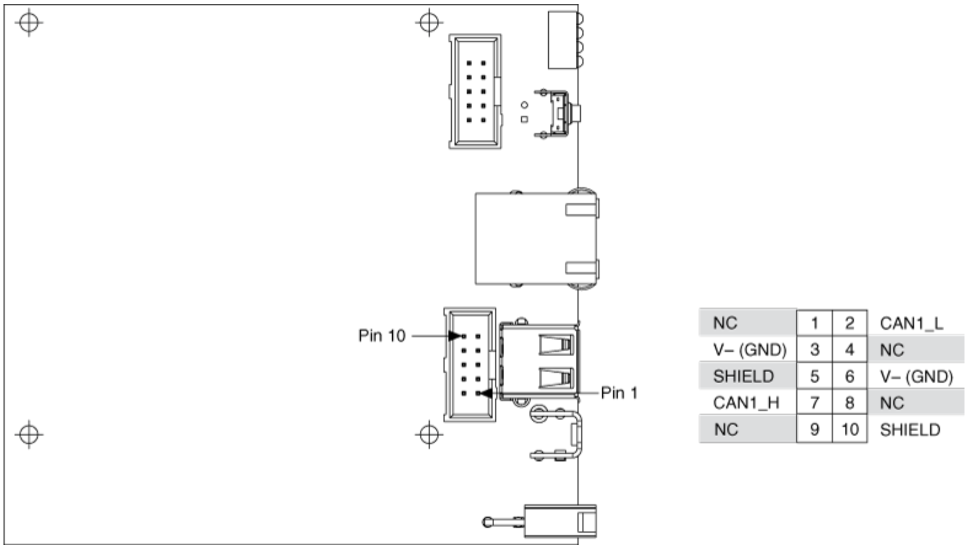
Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9607

Single-Board Reconfigurable I/O (DIO), 1 RS-232 Serial port, 1 CAN port, Zynq-7020 FPGA

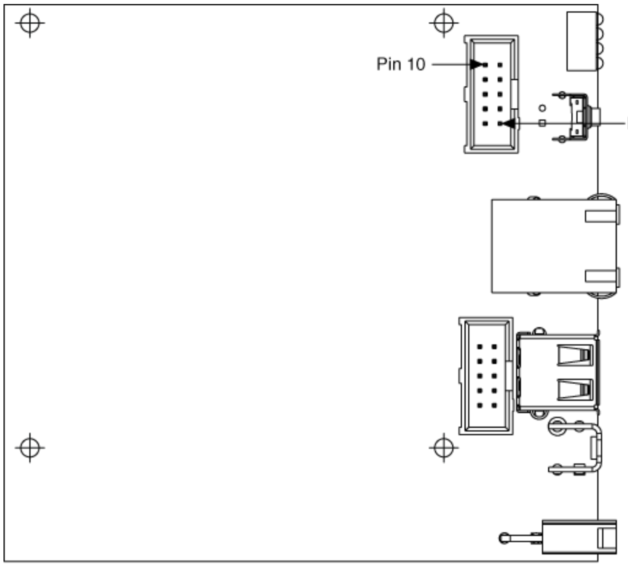
sbRIO-9607 Pinouts

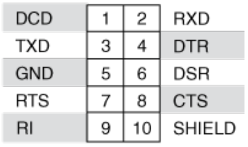
Connector W500, CAN Port



NC	1	2	CAN1_L
V– (GND)	3	4	NC
SHIELD	5	6	V– (GND)
CAN1_H	7	8	NC
NC	9	10	SHIELD

Connector W501, Serial Port





DCD	1	2	RXD
TXD	3	4	DTR
GND	5	6	DSR
RTS	7	8	CTS
RI	9	10	SHIELD

Refer to the CompactRIO Related Documentation for further information about CompactRIO documentation.

Related Topics

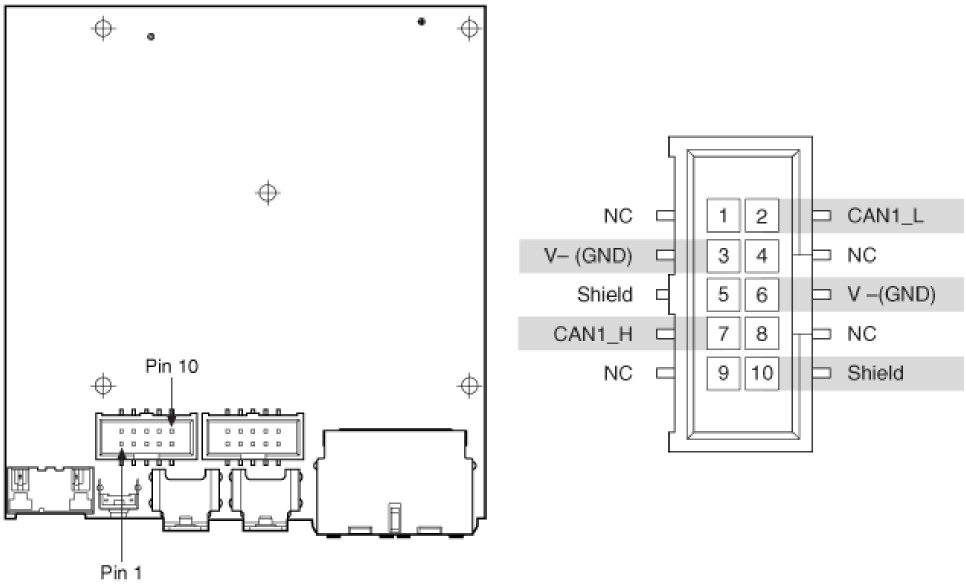
Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9608

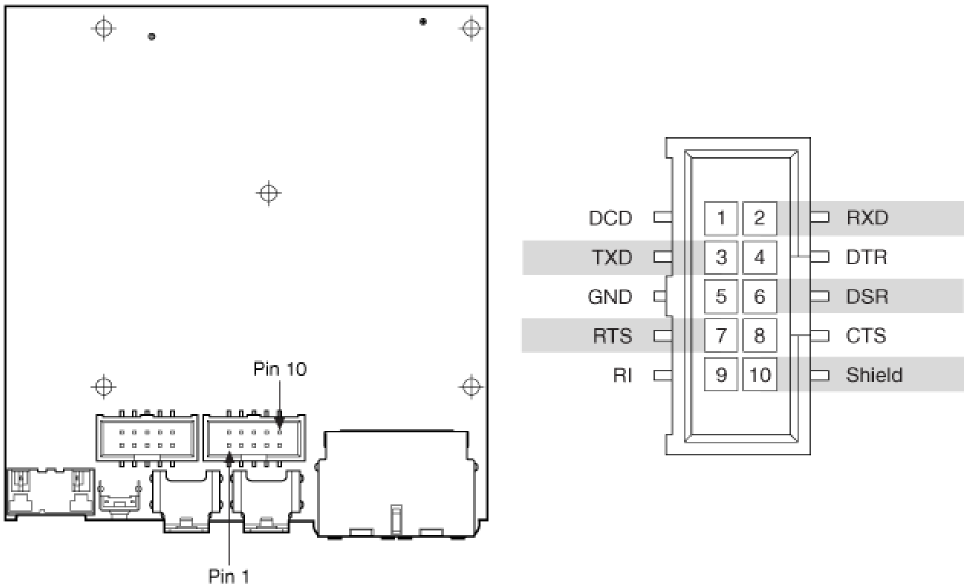
Single-Board Reconfigurable I/O (DIO), 1 RS-232 Serial port, 1 CAN port, Artix-7 200T FPGA

sbRIO-9608 Pinouts

Connector J3, CAN Port



Connector J4, Serial Port



Refer to the CompactRIO Related Documentation for further information about CompactRIO documentation.

Related Topics

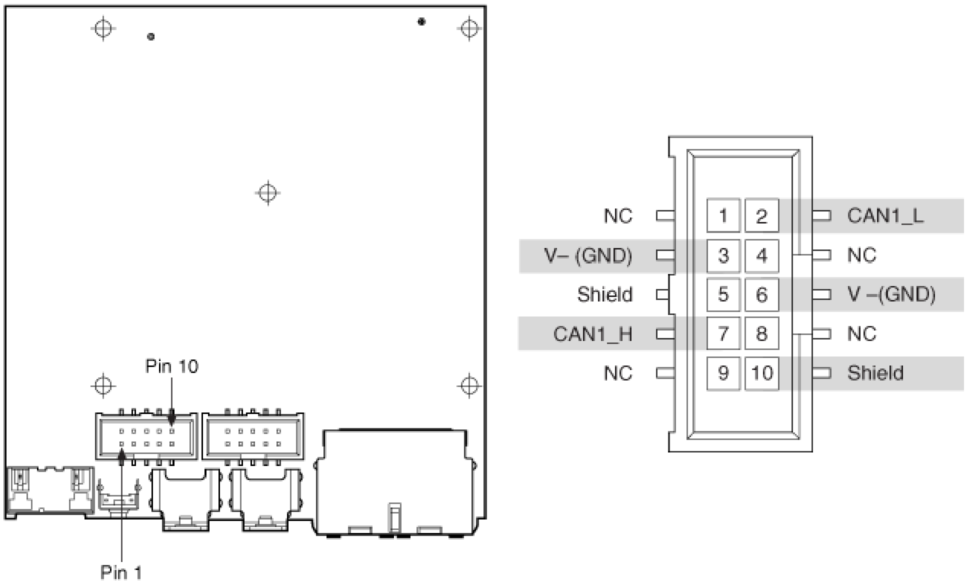
Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9609

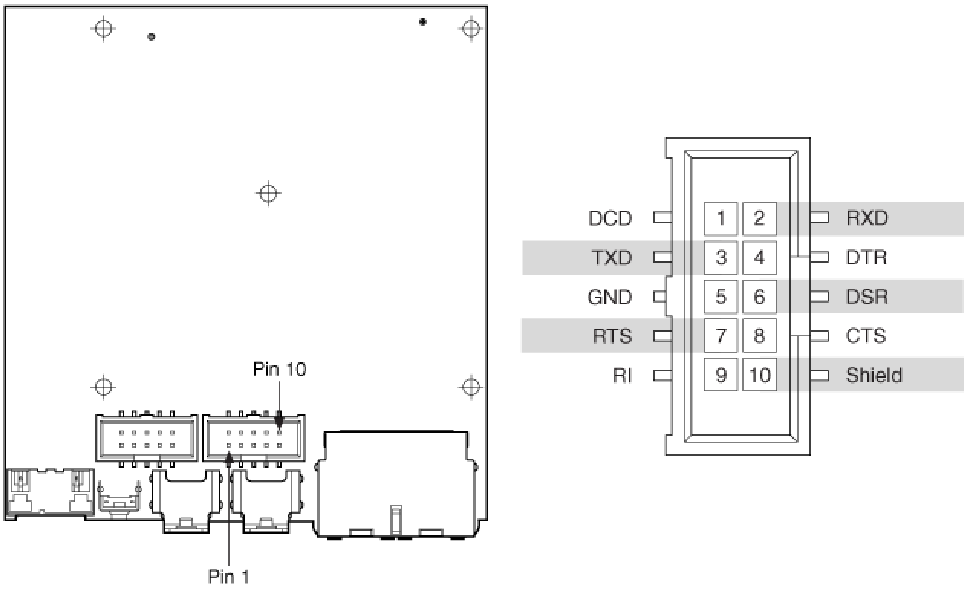
Single-Board Reconfigurable I/O (DIO), 1 RS-232 Serial port, 1 CAN port, Artix-7 200T FPGA

sbRIO-9609 Pinouts

Connector J3, CAN Port



Connector J4, Serial Port



Refer to the CompactRIO Related Documentation for further information about CompactRIO documentation.

Related Topics

Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9627

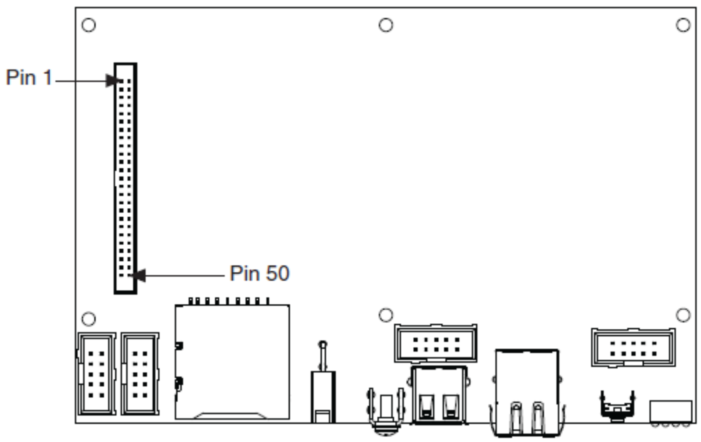
Single-Board Reconfigurable I/O (MIO), 16 AI channels, 4 AO channels, 4 3.3 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, RIO mezzanine card connector, Zynq-7020 FPGA

Software Reference (?)

FPGA Interface

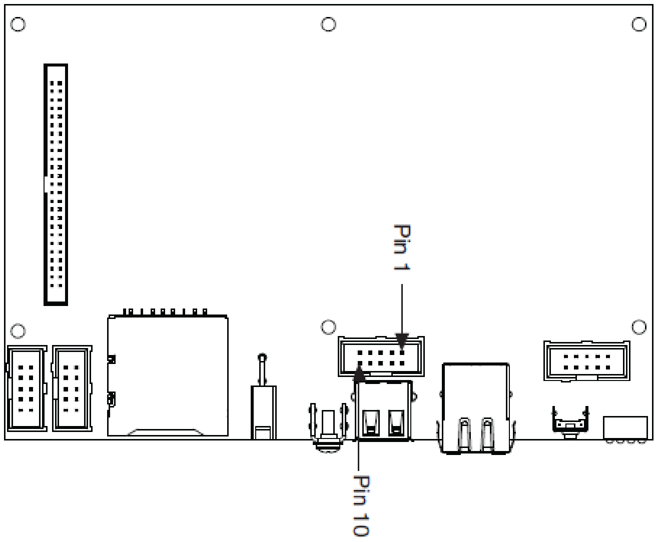
sbRIO-9627 Pinouts

Connector J5, MIO



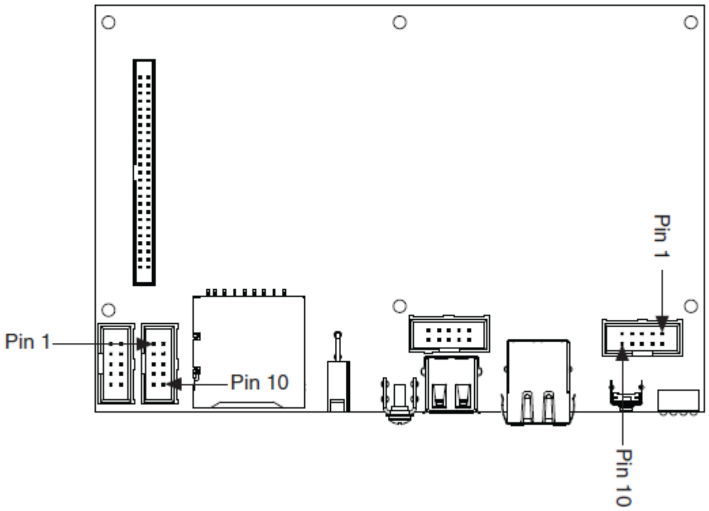
GND	1	2	AI0
AI8	3	4	GND
AI9	5	6	AI1
GND	7	8	AI2
AI10	9	10	GND
AI11	11	12	AI13
GND	13	14	AI14
AI12	15	16	GND
AI13	17	18	AI5
GND	19	20	AI6
AI14	21	22	GND
AI15	23	24	AI7
GND	25	26	GND
GND	27	28	AO0
GND	29	30	AO1
GND	31	32	AO2
GND	33	34	AO3
GND	35	36	NC
GND	37	38	NC
GND	39	40	NC
GND	41	42	NC
GND	43	44	DIO0
GND	45	46	DIO1
GND	47	48	DIO2
GND	49	50	DIO3

Connector W1, CAN Port



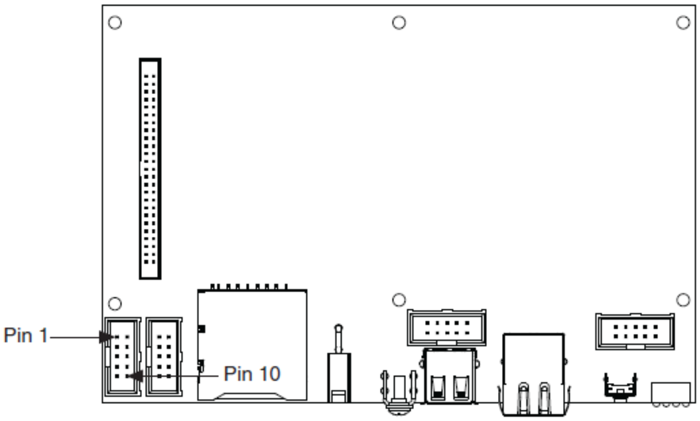
NC	1	2	CAN1_L
V– (GND)	3	4	NC
SHIELD	5	6	V– (GND)
CAN1_H	7	8	NC
NC	9	10	SHIELD

Connectors W2 and W4, RS-232 Serial Ports



DCD	1	2	RXD
TXD	3	4	DTR
GND	5	6	DSR
RTS	7	8	CTS
RI	9	10	SHIELD

Connector W3, RS-485 Serial Port



GND	1	2	NC
NC	3	4	RXD+
RXD–	5	6	NC
NC	7	8	TXD+
TXD–	9	10	SHIELD

Related Topics

Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9627 (FPGA Interface)

Single-Board Reconfigurable I/O (MIO)

16 AI channels, 4 AO channels, 4 3.3 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, RIO mezzanine card connector, Zynq-7020 FPGA

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
Connector0/AI**x**	Analog input channel **x**, where **x** is the number of the channel. The sbRIO-9627 has AI channels 0 to 15.
Connector0/AO**x**	Analog output channel **x**, where **x** is the number of the channel. The sbRIO-9627 has AO channels 0 to 3.
Connector0/DIO**x**	Digital input/output channel **x**, where **x** is the n umber of the channel. The sbRIO-9627 has chan nels 0 to 3. Use the FPGA I/O Node or the Set Ou tput Data or Set Output Enable method to acces s this channel.

Arbitration

You can configure the arbitration settings for the DIO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node topic for a d escription of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a d escription of this method.

Module Methods

This device does not support any module methods.

Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	Description
Terminal Mode	Sets the terminal mode for a channel as RSE (referenced single-ended) or DIFF (differential). This property overwrites the value you configure in the **FPGA I/O Properties** dialog box.
Voltage Range	Sets the input range for a channel as ±10 V, ±5 V, ±2 V, or ±1 V. This property overwrites the value you configure in the **FPGA I/O Properties** dialog box.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop for digital I/O only. Configure the number of output synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. Configure the number of input synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Node Properties dialog box.

sbRIO-9628

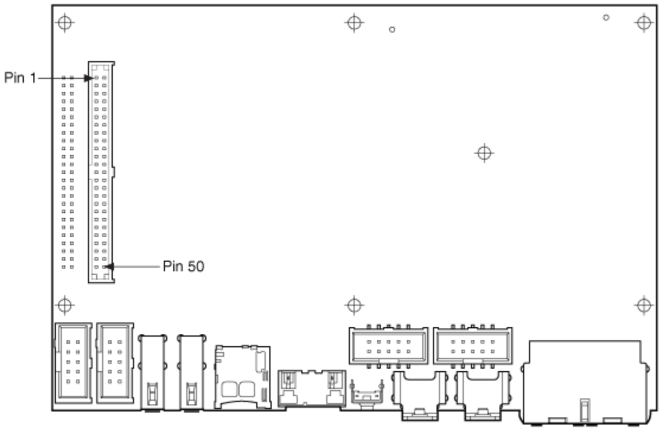
Single-Board Reconfigurable I/O (MIO and DIO), 16 AI channels, 4 AO channels, 4 5 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, RIO mezzanine card connector, Artix-7 100T FPGA

Software Reference (?)

FPGA Interface

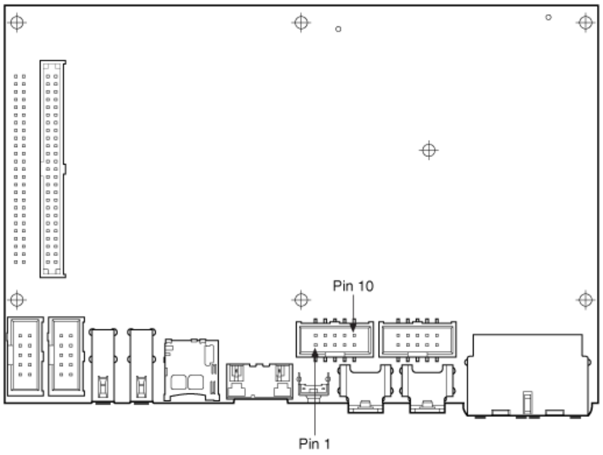
sbRIO-9628 Pinouts

Connector J2, MIO



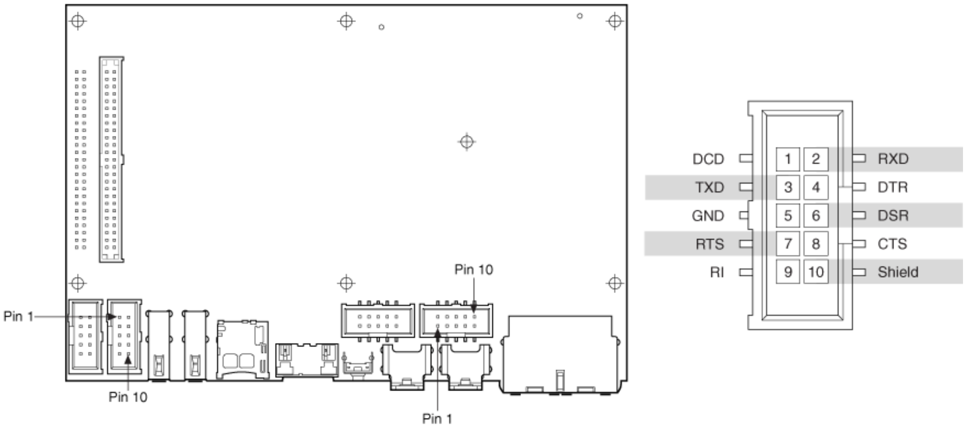
AGND	1	2	AI0
AI8	3	4	AGND
AI9	5	6	AI1
AGND	7	8	AI2
AI10	9	10	AGND
AI11	11	12	AI3
AGND	13	14	AI4
AI12	15	16	AGND
AI13	17	18	AI5
AGND	19	20	AI6
AI14	21	22	AGND
AI15	23	24	AI7
AGND	25	26	AGND
AGND	27	28	AO0
AGND	29	30	AO1
AGND	31	32	AO2
AGND	33	34	AO3
AGND	35	36	NC
AGND	37	38	NC
AGND	39	40	NC
AGND	41	42	NC
DGND	43	44	DIO0
DGND	45	46	DIO1
DGND	47	48	DIO2
DGND	49	50	DIO3

Connector J3, CAN Port

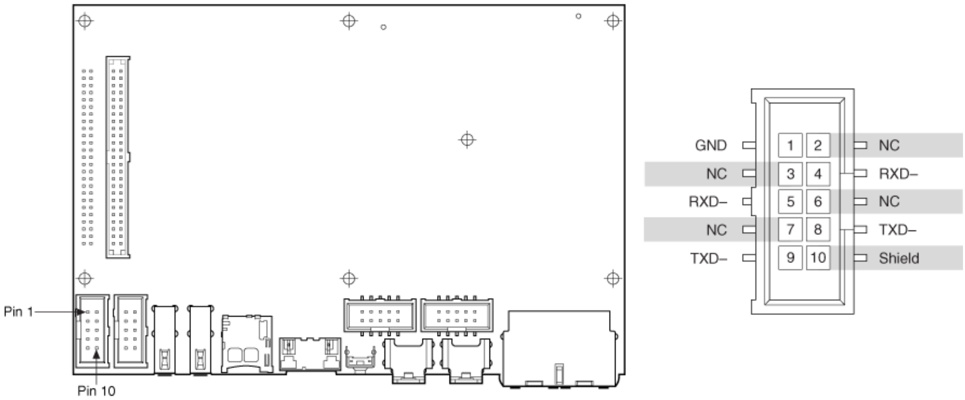


NC	1	2	CAN1_L
V– (GND)	3	4	NC
Shield	5	6	V –(GND)
CAN1_H	7	8	NC
NC	9	10	Shield

Connectors J4 and J6, RS-232 Serial Ports



Connector J5, RS-485 Serial Port



Related Topics

Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9628 (FPGA Interface)

Single-Board Reconfigurable I/O (MIO and DIO), 16 AI channels, 4 AO channels, 4 5 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, RIO mezzanine card connector, Artix-7 100T FPGA

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
Conn0_AI/AI**x**	Analog input channel **x**, where **x** is the number of the channel. The sbRIO-9628 has AI channels 0 to 15.
Conn0_AO/AO**x**	Analog output channel **x**, where **x** is the number of the channel. The sbRIO-9628 has AO channels 0 to 3.
Conn0_DIO0-3/DIO**x**	Digital input/output channel **x**, where **x** is the ch annel number. The sbRIO-9628 has DIO channel s 0 to 3.
Conn0_DIO0-3/DIO3:0	Digital port consisting of channels 0 through 3. Channel 3 signifies the MSB and channel 0 signif ies the LSB.

Arbitration

You cannot configure arbitration settings for analog input and analog output channels of this device. Analog input and analog output channels on this device only support the Arbitrate if Multiple Requestors Only option for arbitration.

You can configure the arbitration settings for digital output channels on this device in the Advanced Code Generation page on the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

Digital input channels on this device only support the Never Arbitrate option for arbitration. You cannot configure arbitration settings for digital input channels on this device.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	I/O Type	Description
Set Output Data	DIO	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	DIO	Sets the line direction of the digital channel or the digital port. Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	I/O Type	Description
Check Status	DIO	Returns a Boolean value that indicates whether the DIO module is ready for I/O operations.

I/O Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	I/O Type	Description
Terminal Mode	AI	Sets the terminal mode for a channel as RSE (referenced single-ended) or DIFF (differential). This property overwrites the value you configure in the **Module Properties** dialog box. You cannot configure channels 8 through 15 to DIFF mode.
Voltage Range	AI	Sets the input range for a channel as ±10 V, ±5 V, ±2 V, or ±1 V. This property overwrites the value you configure in the **Module Properties** dialog box.
LSB Weight	AO	Returns the LSB weight in nV/LSB for the channel. Use this value to convert AO data if you set t

		he **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset	AO	Returns the calibration offset in nV for the channel. Use this value to convert AO data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Module Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	I/O Type	Description
LSB Weight (±10 V range)	AI	Returns the LSB weight in pV/LSB for the ±10 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±5 V range)	AI	Returns the LSB weight in pV/LSB for the ±5 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±2 V range)	AI	Returns the LSB weight in pV/LSB for the ±2 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±1 V range)	AI	Returns the LSB weight in pV/LSB for the ±1 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Offset (±10 V range)	AI	Returns the calibration offset in nV for the ±10 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±5 V range)	AI	Returns the calibration offset in nV for the ±5 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±2 V range)	AI	Returns the calibration offset in nV for the ±2 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±1 V range)	AI	Returns the calibration offset in nV for the ±1 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop for digital I/O only. Configure the number of output synchronizing registers or input synchronizing registers for the channels on this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box.

The Set Output Enable method node is not supported in the single-cycle Timed Loop. When writing to a DIO I/O node in a single-cycle Timed Loop, the I/O node will not change the DIO line directions. To set the line directions to output when writing to the DIO channels in a single-cycle Timed Loop, either configure the default line directions to output in the **Module Properties** dialog box, or use the Set Output Enable method node outside a single-cycle Timed Loop.

This device supports the Number of Synchronizing Registers for Output Data synchronizing register option when used in SCTL output. This option supports the same functionality as the **Number of Synchronizing Registers for Output Data** option described in the Advanced Code Generation FPGA I/O Properties Page (FPGA Module) topic, with the exception that you can use this option only in a single-cycle Timed Loop. You can implement either 0 or 1 synchronizing registers inside the single-cycle Timed Loop, however, if you configure 0 synchronizing registers outside of the single-cycle Timed Loop, the FPGA VI implements 1 synchronizing register by default.

When the device is within a single-cycle Timed Loop, it must be ready to perform digital I/O before a loop containing digital I/O starts. Poll the **Ready** output of the Check Status method to determine whether the module is ready. Digital input operations return invalid data if the module is not ready. The module also might ignore or delay digital output operations if it is not ready.

Converting Analog Input Values for the sbRIO-9628/9629/9638 (FPGA Interface)

Set the **Calibration Mode** to **Calibrated** in the **Module Properties** dialog box for the onboard analog input module if you want the FPGA I/O Node to return calibrated, fixed-point data for the module. If you set the **Calibration Mode** to **Raw**, the FPGA I/O Node returns uncalibrated, binary values. These binary values can be converted to calibrated volts by applying the calibration constants. The conversion of binary data to calibrated data can be done in the host VI.

Using an Equation to Convert and Calibrate Values

You can use the following equation in the host VI to convert binary analog input values to calibrated volts:

Volts = (**Binary Value** × **LSB Weight** - **Offset**)

where

Binary Value is the signed value returned by the FPGA I/O Node

LSB Weight is the value returned by the LSB Weight property

Offset is the value returned by the Offset property.

The **LSB Weight** property is returned in units of pV/LSB. The **Offset** property is returned in units of nV.

To convert to calibrated volts, use the FPGA I/O Property Node to read the LSB Weight and Offset properties for the desired input range. If you do not want to read the LSB Weight and Offset values from the module, you can convert to uncalibrated engineering units by using the following typical values for **Offset** and **LSB Weight**:

Offset = 0 V

The typical **LSB Weight** depends on the input range. See the table below:

Input Range	Typical LSB Weight
±10 V	320 µV/LSB
±5 V	160 µV/LSB
±2 V	64 µV/LSB
±1 V	32 µV/LSB

Converting Analog Output Values for the sbRIO-9628/9629/9638 (FPGA Interface)

Set the **Calibration Mode** to **Calibrated** in the **Module Properties** dialog box for the onboard analog output module if you want the FPGA I/O Node to accept fixed-point data in units of volts for the module. If you set the **Calibration Mode** to **Raw**, the FPGA I/O Node accepts only binary values. You must convert output voltage values to binary values using the calibration constants before you write them to the FPGA I/O node. The conversion of voltage values to binary values can be done in the host VI.

Using an Equation to Convert Voltage to Binary

You can use the following equation in the host VI to convert the analog output values to binary values:

Binary Value = (**Voltage Value** - **Offset**) ÷ **LSB Weight**

where

Binary Value is the value you write to the FPGA I/O Node

Voltage Value is the voltage in V that you want the channel to output

Offset is the value returned by the Offset property

LSB Weight is the value returned by the LSB Weight property.

The **LSB Weight** property is returned in units of nV/LSB. The **Offset** property is returned in units of nV.

NI recommends using calibrated values for analog output. To convert calibrated voltage values, use the FPGA I/O Property Node to read the LSB Weight and Offset properties. If you do not want to read the LSB Weight and Offset values from the module, you can convert uncalibrated voltage values by using the following typical values for **Offset** and **LSB Weight**:

Offset = 0 V

LSB Weight = 320 µV/LSB

Module Properties Dialog Box for the sbRIO-9628/9629/9638 (FPGA Interface)

Right-click an onboard I/O module in the **Project Explorer** window and select **Properties** from the shortcut menu to display this dialog box.

Use this dialog box to configure an onboard I/O module.

For all onboard I/O modules, this dialog box includes the following components:

- **Name**—Specifies the name of the onboard I/O module, which appears in the **Project Explorer** window. LabVIEW assigns a default name to the module based on the connector number and channel numbers. You can use this field to give the onboard module a descriptive name.
- **Module Type**—Specifies the type of module. You cannot change this option.
- **Location**—Specifies the connector containing the onboard module's channels. You cannot change this option.

The onboard AI module's dialog box also includes the following:

- **Calibration Mode**—Sets the calibration mode for the onboard module. Select **Calibrated** if you want the FPGA I/O Node to return calibrated, fixed-point data for the module in units of volts. The fixed-point data is signed, with a word length of 24 bits and an integer word length of 5 bits. Select **Raw** if you want the FPGA I/O Node to return uncalibrated, binary data for the module. If you select **Raw**, you must convert and calibrate the analog input values in the host VI. The default is **Calibrated**.
- **Channel Configuration**—Specifies the input range for each channel.
  - **Channels**—Specifies the channel(s) for which you want to select the input range.
  - **Input Range**—Specifies the input range for the selected channel(s) as ±10 V, ±5 V, ±2 V, or ±1 V.
  - **Terminal Mode**—Specifies the terminal mode for the selected channel(s) as RSE (referenced single-ended) or DIFF (differential).
- **Minimum Time Between Conversions**—Specifies the minimum time between conversions in µs. This time determines the shortest possible time between any two conversions. For channels sampled within the same FPGA I/O Node, the time you set determines the exact time between conversions. For channels sampled within separate FPGA I/O Nodes or for conversions

caused by looping on an FPGA I/O Node, the time you set may be less than the actual time between conversions. However, the minimum time you set is never greater than the time between conversions. If the application tries to execute an FPGA I/O Node faster than the specified minimum time between conversions, the conversion is delayed until the minimum time you set is satisfied. The default minimum time between conversions is 8 µs. The accuracy specifications in the sbRIO-9638 hardware documentation on ni.com/manuals are based on this default value. If you set the minimum time between conversions to at least 8 µs, the accuracy of the module is not affected. If you set the minimum time between conversions to less than 8 µs, the accuracy of the onboard module degrades if you sample data from multiple channels.

The onboard AO module's dialog box also includes the following:

- **Calibration Mode**—Sets the calibration mode for the onboard module. Select **Calibrated** if you want the FPGA I/O Node to accept fixed-point data in units of volts when writing to the module. The fixed-point data is signed, with a word length of 20 bits and an integer word length of 5 bits. Select **Raw** if you want the FPGA I/O Node to accept calibrated, binary data when writing to the module. If you select **Raw**, you must convert the analog output values in the host VI before you write them to the module. The default is **Calibrated**.

The onboard DIO module's dialog box also includes the following:

- **Channels**—Specifies the channel(s) for which you want to select the direction.
- **Selected Channel(s) Settings**—Specifies the default line direction for each channel.

sbRIO-9629

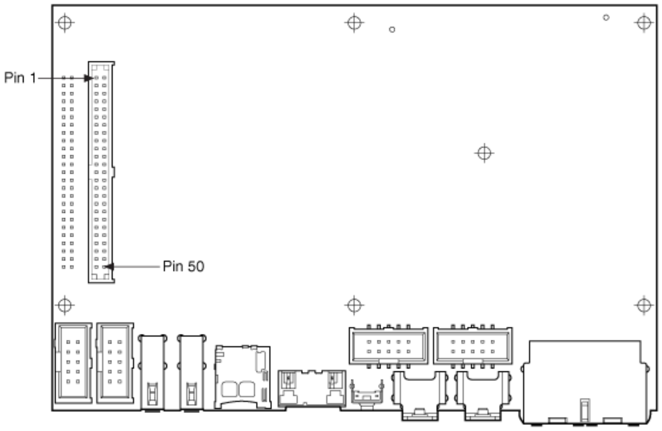
Single-Board Reconfigurable I/O (MIO and DIO), 16 AI channels, 4 AO channels, 4 5 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, RIO mezzanine card connector, Artix-7 200T FPGA

Software Reference (?)

FPGA Interface

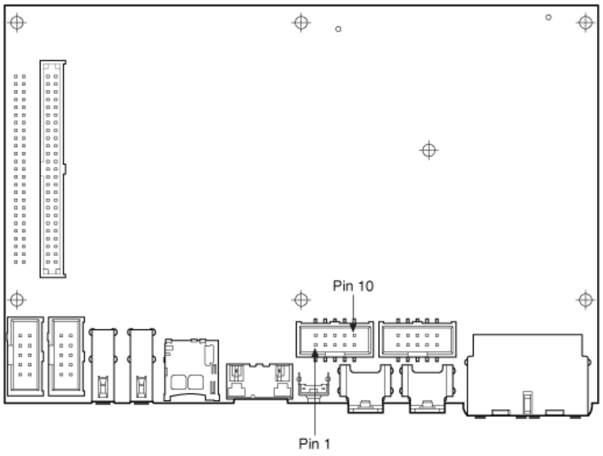
sbRIO-9629 Pinouts

Connector J2, MIO



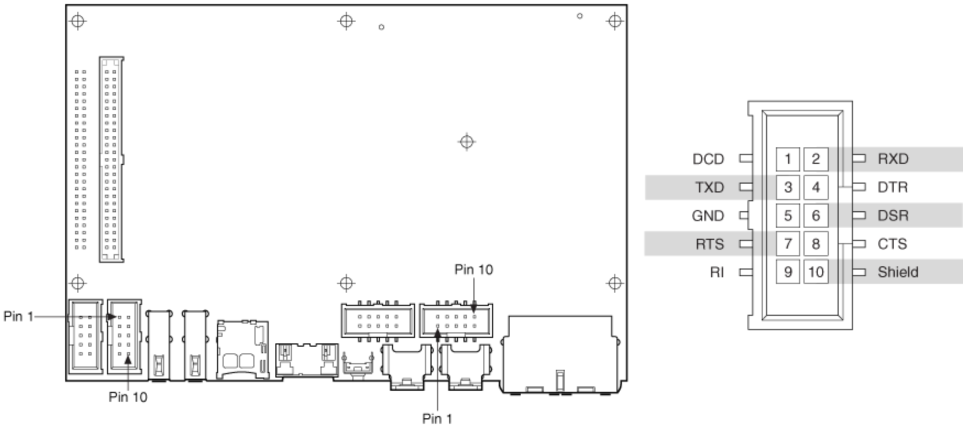
Signal	Pin	Pin	Signal
AGND	1	2	AI0
AI8	3	4	AGND
AI9	5	6	AI1
AGND	7	8	AI2
AI10	9	10	AGND
AI11	11	12	AI3
AGND	13	14	AI4
AI12	15	16	AGND
AI13	17	18	AI5
AGND	19	20	AI6
AI14	21	22	AGND
AI15	23	24	AI7
AGND	25	26	AGND
AGND	27	28	AO0
AGND	29	30	AO1
AGND	31	32	AO2
AGND	33	34	AO3
AGND	35	36	NC
AGND	37	38	NC
AGND	39	40	NC
AGND	41	42	NC
DGND	43	44	DIO0
DGND	45	46	DIO1
DGND	47	48	DIO2
DGND	49	50	DIO3

Connector J3, CAN Port

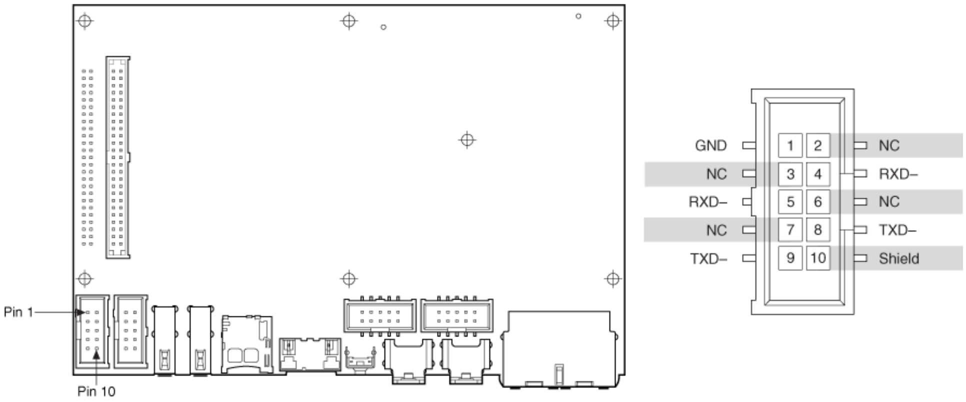


Signal	Pin	Pin	Signal
NC	1	2	CAN1_L
V– (GND)	3	4	NC
Shield	5	6	V –(GND)
CAN1_H	7	8	NC
NC	9	10	Shield

Connectors J4 and J6, RS-232 Serial Ports



Connector J5, RS-485 Serial Port



Related Topics

Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9629 (FPGA Interface)

Single-Board Reconfigurable I/O (MIO and DIO), 16 AI channels, 4 AO channels, 4 5 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, RIO mezzanine card connector, Artix-7 200T FPGA

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
Conn0_AI/AI**x**	Analog input channel **x**, where **x** is the number of the channel. The sbRIO-9629 has AI channels 0 to 15.
Conn0_AO/AO**x**	Analog output channel **x**, where **x** is the number of the channel. The sbRIO-9629 has AO channels 0 to 3.
Conn0_DIO0-3/DIO**x**	Digital input/output channel **x**, where **x** is the ch annel number. The sbRIO-9629 has DIO channel s 0 to 3.
Conn0_DIO0-3/DIO3:0	Digital port consisting of channels 0 through 3. Channel 3 signifies the MSB and channel 0 signif ies the LSB.

Arbitration

You cannot configure arbitration settings for analog input and analog output channels of this device. Analog input and analog output channels on this device only support the Arbitrate if Multiple Requestors Only option for arbitration.

You can configure the arbitration settings for digital output channels on this device in the Advanced Code Generation page on the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

Digital input channels on this device only support the Never Arbitrate option for arbitration. You cannot configure arbitration settings for digital input channels on this device.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	I/O Type	Description
Set Output Data	DIO	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	DIO	Sets the line direction of the digital channel or the digital port. Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	I/O Type	Description
Check Status	DIO	Returns a Boolean value that indicates whether the DIO module is ready for I/O operations.

I/O Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	I/O Type	Description
Terminal Mode	AI	Sets the terminal mode for a channel as RSE (referenced single-ended) or DIFF (differential). This property overwrites the value you configure in the **Module Properties** dialog box. You cannot configure channels 8 through 15 to DIFF mode.
Voltage Range	AI	Sets the input range for a channel as ±10 V, ±5 V, ±2 V, or ±1 V. This property overwrites the value you configure in the **Module Properties** dialog box.
LSB Weight	AO	Returns the LSB weight in nV/LSB for the channel. Use this value to convert AO data if you set t

		he **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset	AO	Returns the calibration offset in nV for the channel. Use this value to convert AO data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Module Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	I/O Type	Description
LSB Weight (±10 V range)	AI	Returns the LSB weight in pV/LSB for the ±10 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±5 V range)	AI	Returns the LSB weight in pV/LSB for the ±5 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±2 V range)	AI	Returns the LSB weight in pV/LSB for the ±2 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±1 V range)	AI	Returns the LSB weight in pV/LSB for the ±1 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Offset (±10 V range)	AI	Returns the calibration offset in nV for the ±10 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±5 V range)	AI	Returns the calibration offset in nV for the ±5 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±2 V range)	AI	Returns the calibration offset in nV for the ±2 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±1 V range)	AI	Returns the calibration offset in nV for the ±1 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop for digital I/O only. Configure the number of output synchronizing registers or input synchronizing registers for the channels on this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box.

The Set Output Enable method node is not supported in the single-cycle Timed Loop. When writing to a DIO I/O node in a single-cycle Timed Loop, the I/O node will not change the DIO line directions. To set the line directions to output when writing to the DIO channels in a single-cycle Timed Loop, either configure the default line directions to output in the **Module Properties** dialog box, or use the Set Output Enable method node outside a single-cycle Timed Loop.

This device supports the Number of Synchronizing Registers for Output Data synchronizing register option when used in SCTL output. This option supports the same functionality as the **Number of Synchronizing Registers for Output Data** option described in the Advanced Code Generation FPGA I/O Properties Page (FPGA Module) topic, with the exception that you can use this option only in a single-cycle Timed Loop. You can implement either 0 or 1 synchronizing registers inside the single-cycle Timed Loop, however, if you configure 0 synchronizing registers outside of the single-cycle Timed Loop, the FPGA VI implements 1 synchronizing register by default.

When the device is within a single-cycle Timed Loop, it must be ready to perform digital I/O before a loop containing digital I/O starts. Poll the **Ready** output of the Check Status method to determine whether the module is ready. Digital input operations return invalid data if the module is not ready. The module also might ignore or delay digital output operations if it is not ready.

sbRIO-9637

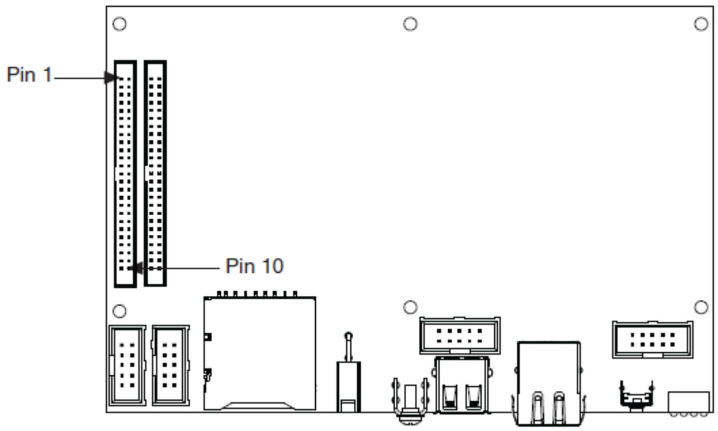
Single-Board Reconfigurable I/O (MIO and DIO), 16 AI channels, 4 AO channels, 28 3.3 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, Zynq-7020 FPGA

Software Reference (?)

FPGA Interface

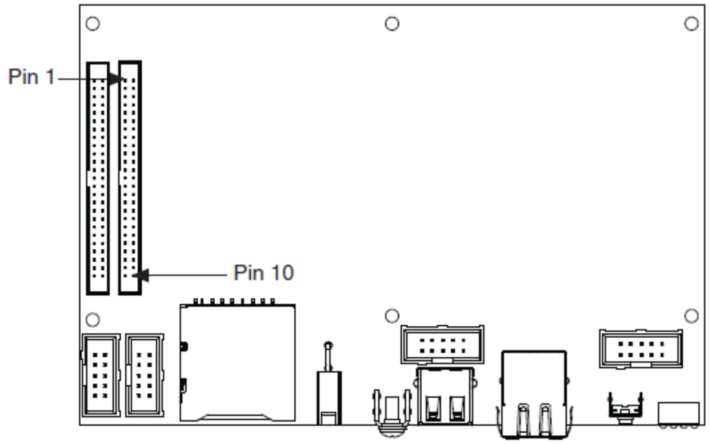
sbRIO-9637 Pinouts

Connector J4, DIO



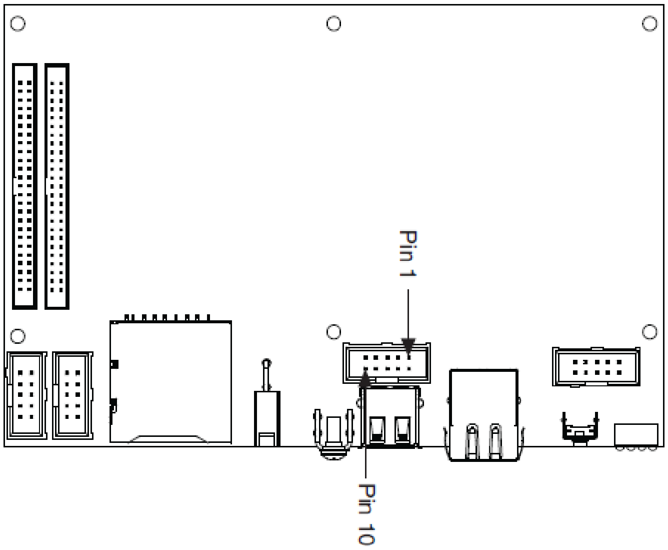
GND	1	2	DIO4
GND	3	4	DIO5
GND	5	6	DIO6
GND	7	8	DIO7
GND	9	10	DIO8
GND	11	12	DIO9
GND	13	14	DIO10
GND	15	16	DIO11
GND	17	18	DIO12
GND	19	20	DIO13
GND	21	22	DIO14
GND	23	24	DIO15
GND	25	26	DIO16
GND	27	28	DIO17
GND	29	30	DIO18
GND	31	32	DIO19
GND	33	34	DIO20
GND	35	36	DIO21
GND	37	38	DIO22
GND	39	40	DIO23
GND	41	42	DIO24
GND	43	44	DIO25
GND	45	46	DIO26
GND	47	48	DIO27
5 V	49	50	5 V

Connector J5, MIO



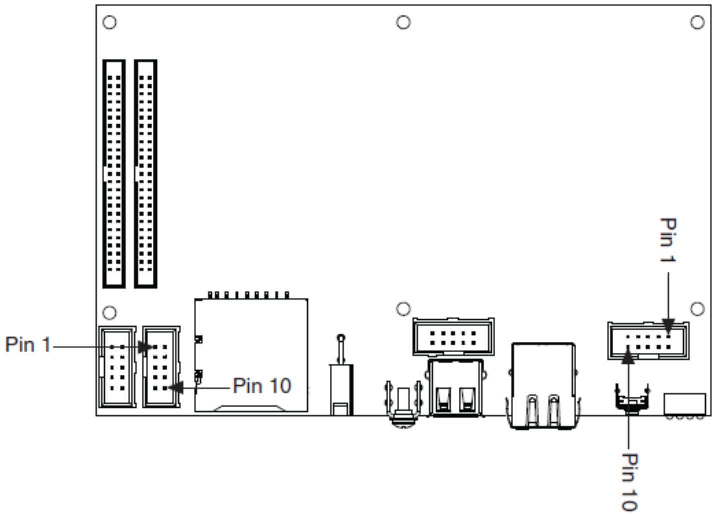
GND	1	2	AI0
AI8	3	4	GND
AI9	5	6	AI1
GND	7	8	AI2
AI10	9	10	GND
AI11	11	12	AI13
GND	13	14	AI14
AI12	15	16	GND
AI13	17	18	AI5
GND	19	20	AI6
AI14	21	22	GND
AI15	23	24	AI7
GND	25	26	GND
GND	27	28	AO0
GND	29	30	AO1
GND	31	32	AO2
GND	33	34	AO3
GND	35	36	NC
GND	37	38	NC
GND	39	40	NC
GND	41	42	NC
GND	43	44	DIO0
GND	45	46	DIO1
GND	47	48	DIO2
GND	49	50	DIO3

Connector W1, CAN Port



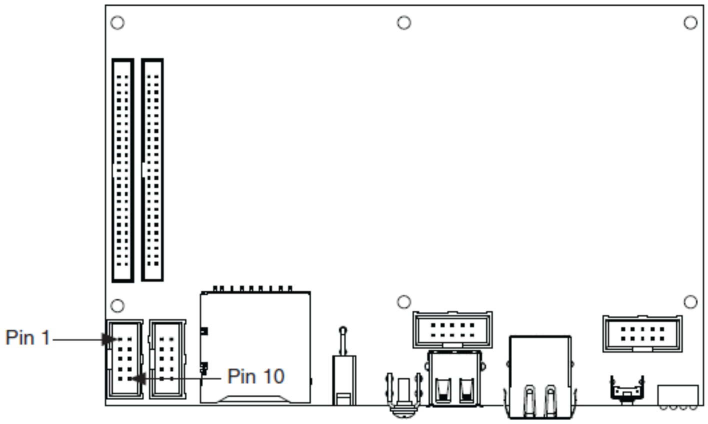
NC	1	2	CAN1_L
V– (GND)	3	4	NC
SHIELD	5	6	V– (GND)
CAN1_H	7	8	NC
NC	9	10	SHIELD

Connectors W2 and W4, RS-232 Serial Ports



DCD	1	2	RXD
TXD	3	4	DTR
GND	5	6	DSR
RTS	7	8	CTS
RI	9	10	SHIELD

Connector W3, RS-485 Serial Port



GND	1	2	NC
NC	3	4	RXD+
RXD–	5	6	NC
NC	7	8	TXD+
TXD–	9	10	SHIELD

Related Topics

Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9637 (FPGA Interface)

Single-Board Reconfigurable I/O (MIO and DIO)

16 AI channels, 4 AO channels, 28 3.3 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, Zynq-7020 FPGA

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
Connector0/AI**x**	Analog input channel **x**, where **x** is the number of the channel. The sbRIO-9637 has AI channels 0 to 15.

Connector0/AO**x**	Analog output channel **x**, where **x** is the number of the channel. The sbRIO-9637 has AO channels 0 to 3.
Connector**x**/DIO**y**	Digital input/output channel **y** on connector **x**, where **y** is the channel number and **x** is the connector number. The sbRIO-9637 has channels 0 to 27. Use the FPGA I/O Node or the Set Output Data or Set Output Enable method to access this channel.

Arbitration

You can configure the arbitration settings for the DIO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

This device does not support any module methods.

Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	Description
Terminal Mode	Sets the terminal mode for a channel as RSE (referenced single-ended) or DIFF (differential). This property overwrites the value you configure in the **FPGA I/O Properties** dialog box.

Voltage Range	Sets the input range for a channel as ±10 V, ±5 V, ±2 V, or ±1 V. This property overwrites the value you configure in the **FPGA I/O Properties** dialog box.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop for digital I/O only. Configure the number of output synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. Configure the number of input synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Node Properties dialog box.

sbRIO-9638

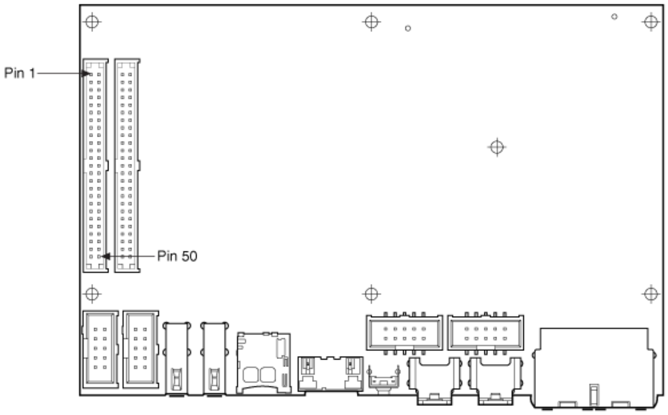
Single-Board Reconfigurable I/O (MIO), 16 AI channels, 4 AO channels, 28 5 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, Artix-7 100T FPGA

Software Reference (?)

FPGA Interface

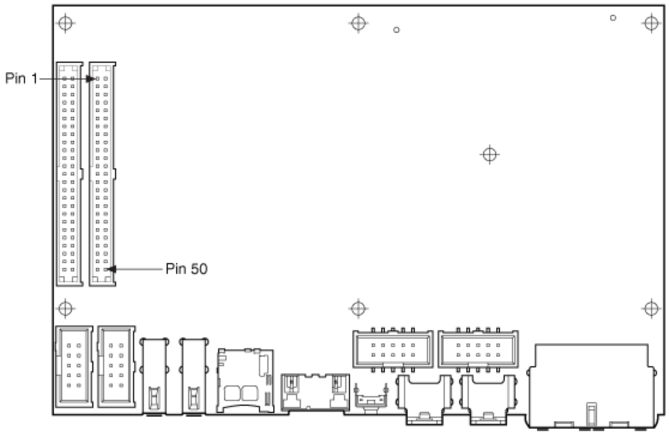
sbRIO-9638 Pinouts

Connector J1, DIO



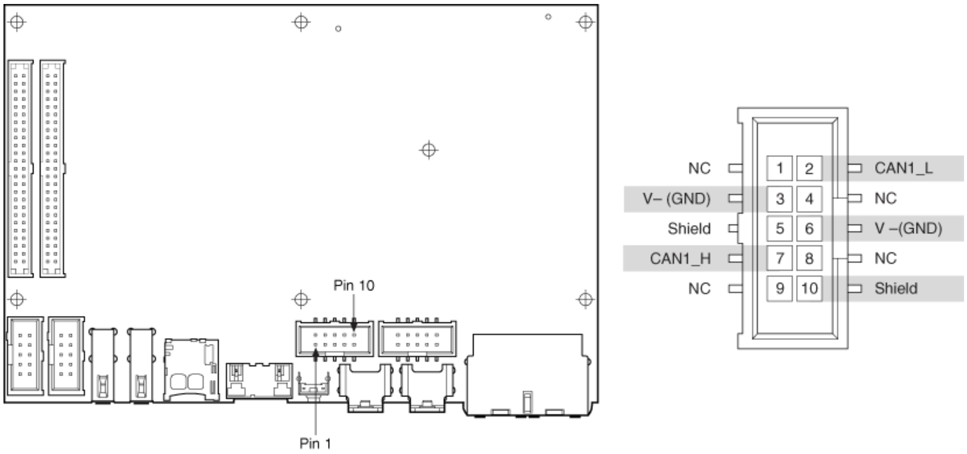
Signal	Pin	Pin	Signal
DGND	1	2	DIO4
DGND	3	4	DIO5
DGND	5	6	DIO6
DGND	7	8	DIO7
DGND	9	10	DIO8
DGND	11	12	DIO9
DGND	13	14	DIO10
DGND	15	16	DIO11
DGND	17	18	DIO12
DGND	19	20	DIO13
DGND	21	22	DIO14
DGND	23	24	DIO15
DGND	25	26	DIO16
DGND	27	28	DIO17
DGND	29	30	DIO18
DGND	31	32	DIO19
DGND	33	34	DIO20
DGND	35	36	DIO21
DGND	37	38	DIO22
DGND	39	40	DIO23
DGND	41	42	DIO24
DGND	43	44	DIO25
DGND	45	46	DIO26
DGND	47	48	DIO27
5 V	49	50	5 V

Connector J2, MIO



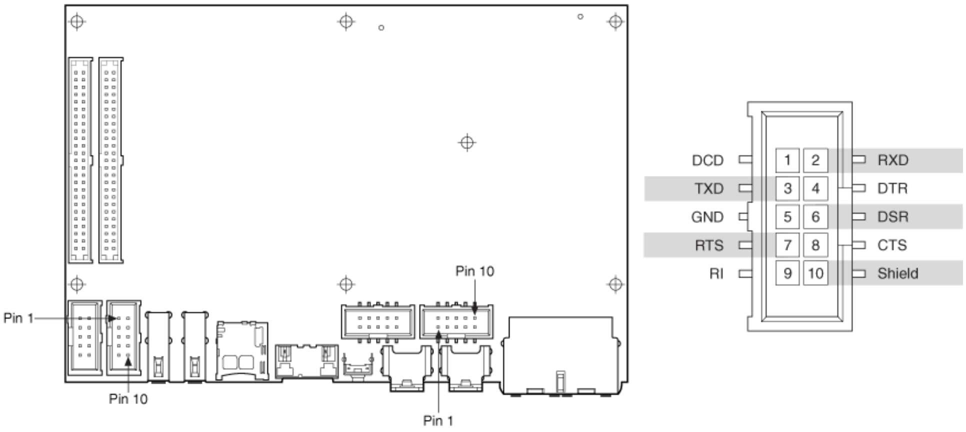
Signal	Pin	Pin	Signal
AGND	1	2	AI0
AI8	3	4	AGND
AI9	5	6	AI1
AGND	7	8	AI2
AI10	9	10	AGND
AI11	11	12	AI3
AGND	13	14	AI4
AI12	15	16	AGND
AI13	17	18	AI5
AGND	19	20	AI6
AI14	21	22	AGND
AI15	23	24	AI7
AGND	25	26	AGND
AGND	27	28	AO0
AGND	29	30	AO1
AGND	31	32	AO2
AGND	33	34	AO3
AGND	35	36	NC
AGND	37	38	NC
AGND	39	40	NC
AGND	41	42	NC
DGND	43	44	DIO0
DGND	45	46	DIO1
DGND	47	48	DIO2
DGND	49	50	DIO3

Connector J3, CAN Port



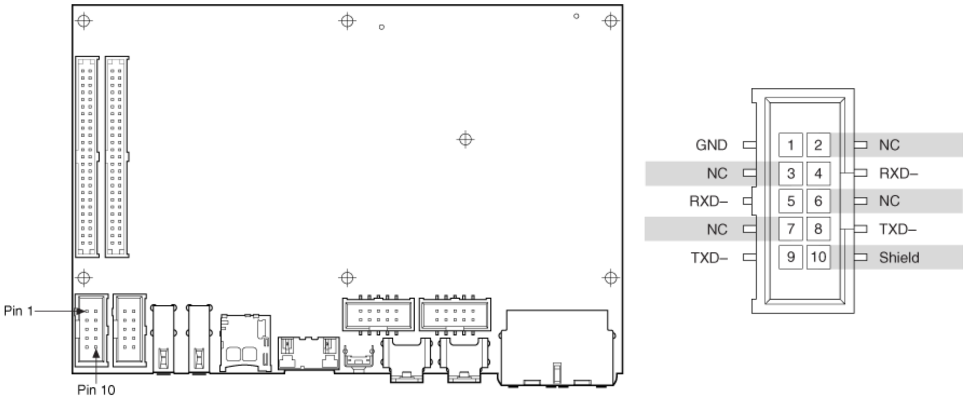
Signal	Pin	Pin	Signal
NC	1	2	CAN1_L
V– (GND)	3	4	NC
Shield	5	6	V –(GND)
CAN1_H	7	8	NC
NC	9	10	Shield

Connectors J4 and J6, RS-232 Serial Ports



Signal	Pin	Pin	Signal
DCD	1	2	RXD
TXD	3	4	DTR
GND	5	6	DSR
RTS	7	8	CTS
RI	9	10	Shield

Connector J5, RS-485 Serial Port



Signal	Pin	Pin	Signal
GND	1	2	NC
NC	3	4	RXD–
RXD–	5	6	NC
NC	7	8	TXD–
TXD–	9	10	Shield

Related Topics

Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

sbRIO-9638 (FPGA Interface)

Single-Board Reconfigurable I/O (MIO), 16 AI channels, 4 AO channels, 28 5 V DIO channels, 2 RS-232 Serial ports, 1 RS-485 Serial port, 1 CAN port, Artix-7 100T FPGA

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
Conn0_AI/AI**x**	Analog input channel **x**, where **x** is the number of the channel. The sbRIO-9638 has AI channels 0 to 15.
Conn0_AO/AO**x**	Analog output channel **x**, where **x** is the number of the channel. The sbRIO-9638 has AO channels 0 to 3.
Conn**w**_DIO**x**-**y**/DIO**z**	Digital input/output channel **z** on connector **w**, where **z** is the channel number, **w** is the connect or number, and **x** and **y** specify the module that contains the channel. The sbRIO-9638 has DIO c hannels 0 to 27 organized into 4 separate modul es.
Conn**w**_DIO**x**-**y**/DIO**y**:**x**	Digital port consisting of channels **x** and **y** on co nnector **w**. Channel **y** signifies the MSB and cha nnel **x** signifies the LSB.

Arbitration

You cannot configure arbitration settings for analog input and analog output channels of this device. Analog input and analog output channels on this device only support the Arbitrate if Multiple Requestors Only option for arbitration.

You can configure the arbitration settings for digital output channels on this device in the Advanced Code Generation page on the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

Digital input channels on this device only support the Never Arbitrate option for arbitration. You cannot configure arbitration settings for digital input channels on this device.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	I/O Type	Description
Set Output Data	DIO	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	DIO	Sets the line direction of the digital channel or the digital port. Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	I/O Type	Description
Check Status	DIO	Returns a Boolean value that indicates whether the DIO module is ready for I/O operations.

I/O Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	I/O Type	Description
Terminal Mode	AI	Sets the terminal mode for a channel as RSE (referenced single-ended) or DIFF (differential). This property overwrites the value you configure in the **Module Properties** dialog box. You cannot configure channels 8 through 15 to DIFF mode.
Voltage Range	AI	Sets the input range for a channel as ±10 V, ±5 V, ±2 V, or ±1 V. This property overwrites the value you configure in the **Module Properties** dialog box.
LSB Weight	AO	Returns the LSB weight in nV/LSB for the channel. Use this value to convert AO data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset	AO	Returns the calibration offset in nV for the channel. Use this value to convert AO data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Module Properties

Use the FPGA I/O Property Node to access the following properties for this device.

Property	I/O Type	Description
LSB Weight (±10 V range)	AI	Returns the LSB weight in pV/LSB for the ±10 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

LSB Weight (±5 V range)	AI	Returns the LSB weight in pV/LSB for the ±5 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±2 V range)	AI	Returns the LSB weight in pV/LSB for the ±2 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
LSB Weight (±1 V range)	AI	Returns the LSB weight in pV/LSB for the ±1 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±10 V range)	AI	Returns the calibration offset in nV for the ±10 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±5 V range)	AI	Returns the calibration offset in nV for the ±5 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±2 V range)	AI	Returns the calibration offset in nV for the ±2 V range. Use this value to convert AI data if you set the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.
Offset (±1 V range)	AI	Returns the calibration offset in nV for the ±1 V range. Use this value to convert AI data if you se

		t the **Calibration Mode** to **Raw** in the **Module Properties** dialog box.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop for digital I/O only. Configure the number of output synchronizing registers or input synchronizing registers for the channels on this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box.

The Set Output Enable method node is not supported in the single-cycle Timed Loop. When writing to a DIO I/O node in a single-cycle Timed Loop, the I/O node will not change the DIO line directions. To set the line directions to output when writing to the DIO channels in a single-cycle Timed Loop, either configure the default line directions to output in the **Module Properties** dialog box, or use the Set Output Enable method node outside a single-cycle Timed Loop.

This device supports the Number of Synchronizing Registers for Output Data synchronizing register option when used in SCTL output. This option supports the same functionality as the **Number of Synchronizing Registers for Output Data** option described in the Advanced Code Generation FPGA I/O Properties Page (FPGA Module) topic, with the exception that you can use this option only in a single-cycle Timed Loop. You can implement either 0 or 1 synchronizing registers inside the single-cycle Timed Loop, however, if you configure 0 synchronizing registers outside of the single-cycle Timed Loop, the FPGA VI implements 1 synchronizing register by default.

When the device is within a single-cycle Timed Loop, it must be ready to perform digital I/O before a loop containing digital I/O starts. Poll the **Ready** output of the Check Status method to determine whether the module is ready. Digital input operations return invalid data if the module is not ready. The module also might ignore or delay digital output operations if it is not ready.

Converting Analog Input Values for the sbRIO-9628/9629/9638 (FPGA Interface)

Set the **Calibration Mode** to **Calibrated** in the **Module Properties** dialog box for the onboard analog input module if you want the FPGA I/O Node to return calibrated, fixed-point data for the module. If you set the **Calibration Mode** to **Raw**, the FPGA I/O Node returns uncalibrated, binary values. These binary values can be converted to calibrated volts by applying the calibration constants. The conversion of binary data to calibrated data can be done in the host VI.

Using an Equation to Convert and Calibrate Values

You can use the following equation in the host VI to convert binary analog input values to calibrated volts:

Volts = (**Binary Value** × **LSB Weight** - **Offset**)

where

Binary Value is the signed value returned by the FPGA I/O Node

LSB Weight is the value returned by the LSB Weight property

Offset is the value returned by the Offset property.

The **LSB Weight** property is returned in units of pV/LSB. The **Offset** property is returned in units of nV.

To convert to calibrated volts, use the FPGA I/O Property Node to read the LSB Weight and Offset properties for the desired input range. If you do not want to read the LSB Weight and Offset values from the module, you can convert to uncalibrated engineering units by using the following typical values for **Offset** and **LSB Weight**:

Offset = 0 V

The typical **LSB Weight** depends on the input range. See the table below:

Input Range	Typical LSB Weight
±10 V	320 µV/LSB
±5 V	160 µV/LSB
±2 V	64 µV/LSB
±1 V	32 µV/LSB

Converting Analog Output Values for the sbRIO-9628/9629/9638 (FPGA Interface)

Set the **Calibration Mode** to **Calibrated** in the **Module Properties** dialog box for the onboard analog output module if you want the FPGA I/O Node to accept fixed-point data in units of volts for the module. If you set the **Calibration Mode** to **Raw**, the FPGA I/O Node accepts only binary values. You must convert output voltage values to binary values using the calibration constants before you write them to the FPGA I/O node. The conversion of voltage values to binary values can be done in the host VI.

Using an Equation to Convert Voltage to Binary

You can use the following equation in the host VI to convert the analog output values to binary values:

Binary Value = (**Voltage Value** - **Offset**) ÷ **LSB Weight**

where

Binary Value is the value you write to the FPGA I/O Node

Voltage Value is the voltage in V that you want the channel to output

Offset is the value returned by the Offset property

LSB Weight is the value returned by the LSB Weight property.

The **LSB Weight** property is returned in units of nV/LSB. The **Offset** property is returned in units of nV.

NI recommends using calibrated values for analog output. To convert calibrated voltage values, use the FPGA I/O Property Node to read the LSB Weight and Offset properties. If you do not want to read the LSB Weight and Offset values from the module, you can convert uncalibrated voltage values by using the following typical values for **Offset** and **LSB Weight**:

Offset = 0 V

LSB Weight = 320 µV/LSB

Module Properties Dialog Box for the sbRIO-9628/9629/9638 (FPGA Interface)

Right-click an onboard I/O module in the **Project Explorer** window and select **Properties** from the shortcut menu to display this dialog box.

Use this dialog box to configure an onboard I/O module.

For all onboard I/O modules, this dialog box includes the following components:

- **Name**—Specifies the name of the onboard I/O module, which appears in the **Project Explorer** window. LabVIEW assigns a default name to the module based on the connector number and channel numbers. You can use this field to give the onboard module a descriptive name.
- **Module Type**—Specifies the type of module. You cannot change this option.
- **Location**—Specifies the connector containing the onboard module's channels. You cannot change this option.

The onboard AI module's dialog box also includes the following:

- **Calibration Mode**—Sets the calibration mode for the onboard module. Select **Calibrated** if you want the FPGA I/O Node to return calibrated, fixed-point data for the module in units of volts. The fixed-point data is signed, with a word length of 24 bits and an integer word length of 5 bits. Select **Raw** if you want the FPGA I/O Node to return uncalibrated, binary data for the module. If you select **Raw**, you must convert and calibrate the analog input values in the host VI. The default is **Calibrated**.

- **Channel Configuration**—Specifies the input range for each channel.
  - **Channels**—Specifies the channel(s) for which you want to select the input range.
  - **Input Range**—Specifies the input range for the selected channel(s) as ±10 V, ±5 V, ±2 V, or ±1 V.
  - **Terminal Mode**—Specifies the terminal mode for the selected channel(s) as RSE (referenced single-ended) or DIFF (differential).
- **Minimum Time Between Conversions**—Specifies the minimum time between conversions in µs. This time determines the shortest possible time between any two conversions. For channels sampled within the same FPGA I/O Node, the time you set determines the exact time between conversions. For channels sampled within separate FPGA I/O Nodes or for conversions caused by looping on an FPGA I/O Node, the time you set may be less than the actual time between conversions. However, the minimum time you set is never greater than the time between conversions. If the application tries to execute an FPGA I/O Node faster than the specified minimum time between conversions, the conversion is delayed until the minimum time you set is satisfied. The default minimum time between conversions is 8 µs. The accuracy specifications in the sbRIO-9638 hardware documentation on ni.com/manuals are based on this default value. If you set the minimum time between conversions to at least 8 µs, the accuracy of the module is not affected. If you set the minimum time between conversions to less than 8 µs, the accuracy of the onboard module degrades if you sample data from multiple channels.

The onboard AO module’s dialog box also includes the following:

- **Calibration Mode**—Sets the calibration mode for the onboard module. Select **Calibrated** if you want the FPGA I/O Node to accept fixed-point data in units of volts when writing to the module. The fixed-point data is signed, with a word length of 20 bits and an integer word length of 5 bits. Select **Raw** if you want the FPGA I/O Node to accept calibrated, binary data when writing to the module. If you select **Raw**, you must convert the analog output values in the host VI before you write them to the module. The default is **Calibrated**.

The onboard DIO module's dialog box also includes the following:

- **Channels**—Specifies the channel(s) for which you want to select the direction.
- **Selected Channel(s) Settings**—Specifies the default line direction for each channel.

sbRIO-9651 SOM

Embedded System on Module (SOM) with Real-Time Processor and Reconfigurable FPGA, 160 configurable DIO channels, 85,000 logic cells

Software Reference (?)

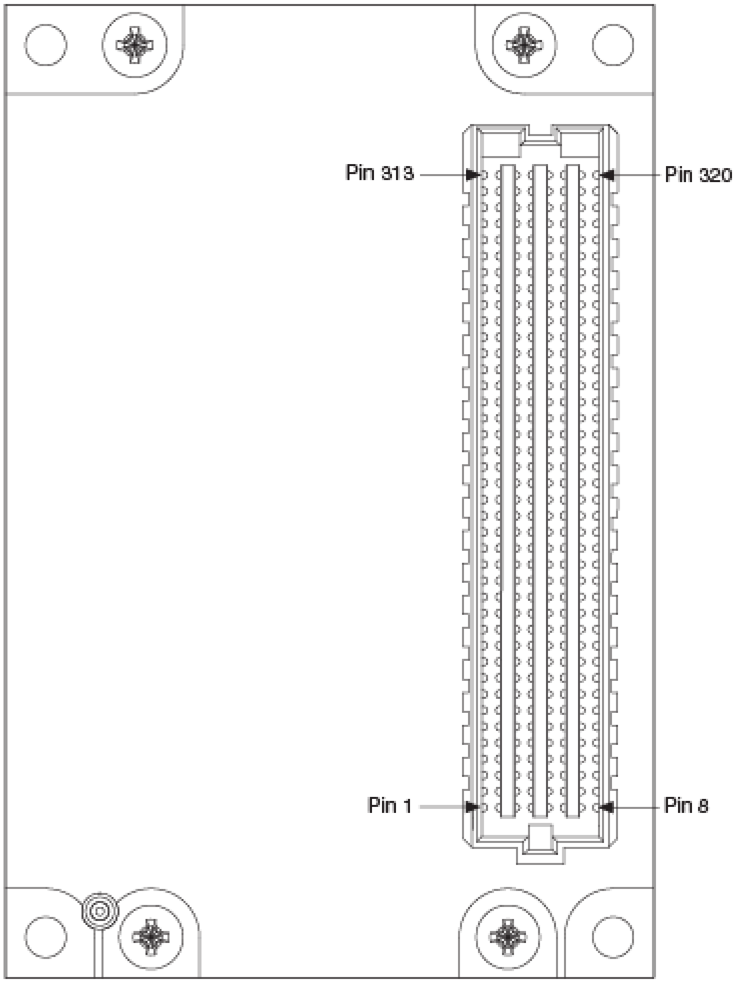
Getting Started with the sbRIO-9651 SOM in LabVIEW

Hardware Documentation (?)

sbRIO-9651 SOM hardware documentation on ni.com/manuals

sbRIO-9651 Pin Assignments

J1 Connector



Legend

USB & Ethernet

LED Signals (LVTTL)

SDIO Interface

Support Signals (LVTTL)

BANK0 (LVTTL)/DIO 0–15

BANK1/DIO 16–39

BANK2/DIO 40–63

BANK3/DIO 64–87

Power In

Ground

313 GND	314 DIO_69	315 GND	316 DIO_75	317 GND	318 DIO_81	319 GND	320 DIO_87_SRCC
305 DIO_66_N	306 DIO_69_N	307 DIO_72_N	308 DIO_75_N	309 DIO_78_N	310 DIO_81_N	311 DIO_84_N	312 DIO_87_N
297 DIO_66	298 GND	299 DIO_72	300 GND	301 DIO_78	302 GND	303 DIO_84_SRCC	304 GND
289 GND	290 DIO_68	291 GND	292 DIO_74	293 GND	294 DIO_80	295 GND	296 DIO_86_MRCC
281 DIO_65_N	282 DIO_68_N	283 DIO_71_N	284 DIO_74_N	285 DIO_77_N	286 DIO_80_N	287 DIO_83_N	288 DIO_86_N
273 DIO_65	274 GND	275 DIO_71	276 GND	277 DIO_77	278 GND	279 DIO_83	280 GND
265 GND	266 DIO_67	267 GND	268 DIO_73	269 GND	270 DIO_79	271 GND	272 DIO_85_MRCC
257 DIO_64_N	258 DIO_67_N	259 DIO_70_N	260 DIO_73_N	261 DIO_76_N	262 DIO_79_N	263 DIO_82_N	264 DIO_85_N
249 DIO_64	250 GND	251 DIO_70	252 GND	253 DIO_76	254 GND	255 DIO_82	256 GND
241 GND	242 DIO_46	243 GND	244 DIO_52	245 GND	246 DIO_58	247 GND	248 VIO_BANK3
233 DIO_42_N	234 DIO_46_N	235 DIO_49_N	236 DIO_52_N	237 DIO_55_N	238 DIO_58_N	239 DIO_61_N	240 VIO_BANK3
225 DIO_52	226 GND	227 DIO_49	228 GND	229 DIO_55	230 GND	231 DIO_61_MRCC	232 GND
217 GND	218 DIO_45	219 GND	220 DIO_51	221 GND	222 DIO_57	223 GND	224 DIO_63_SRCC
209 DIO_41_N	210 DIO_45_N	211 DIO_48_N	212 DIO_51_N	213 DIO_54_N	214 DIO_57_N	215 DIO_60_N	216 DIO_63_N

201 DIO_41	202 GND	203 DIO_48	204 GND	205 DIO_54	206 GND	207 DIO_60_SRCC	208 GND
193 GND	194 DIO_44	195 GND	196 DIO_50	197 GND	198 DIO_56	199 GND	200 DIO_62_MRCC
185 DIO_40_N	186 DIO_44_N	187 DIO_47_N	188 DIO_50_N	189 DIO_53_N	190 DIO_56_N	191 DIO_59_N	192 DIO_62_N
177 DIO_40	178 GND	179 DIO_47	180 GND	181 DIO_53	182 GND	183 DIO_59	184 GND
169 GND	170 DIO_43	171 GND	172 DIO_28	173 GND	174 DIO_35_PUDC	175 GND	176 VIO_BANK2
161 DIO_18_N	162 DIO_43_N	163 DIO_24_N	164 DIO_28_N	165 DIO_31_N	166 DIO_35_N	167 DIO_38_N	168 VIO_BANK2
153 DIO_18	154 GND	155 DIO_24	156 GND	157 DIO_31	158 GND	159 DIO_38_MRCC	160 GND
145 GND	146 DIO_21	147 GND	148 DIO_27	149 GND	150 DIO_34	151 GND	152 DIO_39_SRCC
137 DIO_17_N	138 DIO_21_N	139 DIO_23_N	140 DIO_27_N	141 DIO_30_N	142 DIO_34_N	143 DIO_37_N	144 DIO_39_N
129 DIO_17	130 GND	131 DIO_23	132 GND	133 DIO_30	134 GND	135 DIO_37_MRCC	136 GND
121 GND	122 DIO_20	123 GND	124 DIO_26	125 GND	126 DIO_33	127 GND	128 VIO_BANK1
113 DIO_16_N	114 DIO_20_N	115 DIO_22_N	116 DIO_26_N	117 DIO_29_N	118 DIO_33_N	119 DIO_36_N	120 VIO_BANK1
105 DIO_16	106 GND	107 DIO_22	108 GND	109 DIO_29	110 GND	111 DIO_36_SRCC	112 GND
97 GND	98 DIO_19	99 GND	100 DIO_25	101 GND	102 DIO_32	103 GND	104 VCC_3V3

89 GND	90 DIO_19_N	91 GND	92 DIO_25_N	93 GND	94 DIO_32_N	95 GND	96 VCC_3V3
81 USB0_VBUS	82 GND	83 DIO_4	84 GND	85 DIO_8	86 GND	87 DIO_15_MRCC	88 VCC_3V3
73 USB0_CPEN	74 DIO_1	75 DIO_3	76 DIO_6	77 GND	78 DIO_12	79 DIO_14	80 VCC_3V3
65 USB0_MODE	66 DIO_0	67 GND	68 DIO_5	69 DIO_7	70 DIO_11	71 GND	72 GND
57 GND	58 USB1_VBUS	59 DIO_2	60 Serial1_RX	61 GND	62 DIO_10	63 DIO_13	64 VBAT
49 GND	50 USB1_CPEN	51 GND	52 Serial1_TX	53 FPGA_CFG	54 DIO_9	55 GND	56 RESERVED-NC
41 USB0_DN	42 GND	43 USB1_DN	44 GND	45 GND	46 TEMP_ALERT	47 SYS_RST#	48 GND
33 USB0_DP	34 GND	35 USB1_DP	36 GND	37 CARRIER_RST#	38 SD_PWR_EN	39 GND	40 SD_WP
25 GND	26 GBE0_MDI1_N	27 GND	28 GBE0_MDI3_N	29 GND	30 SD_D3	31 SD_CMD	32 SD_CD#
17 GND	18 GBE0_MDI1_P	19 GND	20 GBE0_MDI3_P	21 GND	22 SD_D1	23 GND	24 SD_D2
9 GBE0_MDI0_N	10 GND	11 GBE0_MDI2_N	12 GND	13 GBE0_SPEED_LEDy	14 STATUS_LED	15 SD_CLK	16 GND
1 GBE0_MDI0_P	2 GND	3 GBE0_MDI2_P	4 GND	5 GBE0_SPEED_LEDg	6 GBE0_ACT_LEDg	7 GND	8 SD_D0

Related Topics

Configuring a Project with Connected Hardware
Configuring a Project with Offline Hardware
RT Watchdog VIs

Getting Started with the sbRIO-9651 in LabVIEW

The sbRIO-9651 System on Module (SOM) requires a mated carrier board in order to function as a complete LabVIEW FPGA and LabVIEW Real-Time target. The sbRIO-9651 reserves some pins on its connector for specific functionality, such as primary Ethernet and USB ports, but provides several banks of additional pins that you can configure for purposes specific to your application.

The sbRIO-9651 SOM Development Kit includes a reference carrier board, which shows an example of how to implement additional I/O functionality you might require, such as Serial, CAN, and secondary Ethernet.

To use the sbRIO-9651 as a LabVIEW FPGA and LabVIEW Real-Time target in a LabVIEW project, you must first create a socketed component-level IP (CLIP) that defines the I/O configuration to use in your application.

Use this tutorial to create a CLIP for the sbRIO-9651 and add it to a LabVIEW project.

Before You Begin

- Before you write any software, use the **sbRIO-9651 System on Module Carrier Board Design Guide** on ni.com/manuals to plan the layout of your carrier board and define the purpose of each pin on the connector. When you create a CLIP for the sbRIO-9651, you must already know the following details:
  - Which processor peripherals you want to enable
  - What pins you want to use
  - What I/O standards and drive strength you want to use for the pins
  - What I/O voltage level you require
  - What LabVIEW FPGA I/O Nodes your application will require
  - What timing constraints or clock resources you require

		Get started: ▶ Creating Component-Level IP for the sbRIO-9651

Related Topics

Using VHDL Code as Component-Level IP (FPGA Module)

Creating Component-Level IP for the sbRIO-9651

Use the sbRIO CLIP Generator to generate a unique socketed CLIP that you can import into a LabVIEW project. The sbRIO CLIP Generator completes the following tasks for you:

- Generates processor peripherals
- Configures the raw FPGA I/O pins
- Generates a LabVIEW FPGA I/O Node interface for the socketed CLIP

Complete the following steps to launch the sbRIO CLIP Generator and create a new CLIP.

1. Create a new LabVIEW project.
2. Add an FPGA target to the project. In the Add Targets and Devices dialog box, select **Real-Time Single-Board RIO»sbRIO-9651**.
3. In the Project Explorer window, right-click the **Chassis (sbRIO-9651)** item and select **New»FPGA Target**.
4. Right-click the **FPGA Target** item and select **Launch sbRIO CLIP Generator**.

Note You can also launch the sbRIO CLIP Generator outside of LabVIEW in the following ways:

- (Windows 8.**x**) Click the **NI Launcher** tile on the Start screen and select **CompactRIO»Single-Board RIO»sbRIO CLIP Generator** .

- (Windows 7 or earlier) Select **Start»All Programs»NI»CompactRIO»Single-Board RIO»sbRIO CLIP Generator** .

5. Step through the pages of the sbRIO CLIP Generator to name the CLIP, configure the peripherals you want to enable, create a LabVIEW FPGA I/O Node interface for the CLIP, and configure clock resources for the CLIP.

Tip Click **Help** in the sbRIO CLIP Generator or browse to the `<NI>\CompactRIO\sbRIO\CLIP Generator` directory to launch the **NI Single-Board RIO CLIP Generator Help**, which contains more information about using the sbRIO CLIP Generator.

Note If you want to make changes to an existing CLIP after it has been created, you must manually edit the generated CLIP files. You cannot use the sbRIO CLIP Generator to edit an existing CLIP. However, you can load the configuration for a previously saved CLIP so you can use it as the basis for a new CLIP.

◀ **Review:**		**Continue:** ▶
Getting Started with the sbRIO-9651 in LabVIEW		Adding a CLIP for the sbRIO-9651 to a LabVIEW Project

Related Topics

Using VHDL Code as Component-Level IP (FPGA Module)

Adding a CLIP for a Single-Board RIO Device to a LabVIEW Project

Complete the following steps to add the CLIP you created using the sbRIO CLIP Generator for a Single-Board RIO target to a LabVIEW project.

Before You Begin

- Use the sbRIO CLIP Generator to generate a unique socketed CLIP.

1. Create a new project.
2. Add an FPGA target to the project. In the Add Targets and Devices dialog box, select **Real-Time Single-Board RIO** and then select your device.
3. In the Project Explorer window, right-click the **Chassis** item and select **New»FPGA Target**.
4. Right-click the **RMC Socket** or **sbRIO-9651 Socket** item and select **Properties** to launch the Component-Level IP Properties or **RMC Socket Properties** dialog box.
5. On the General page, use the **RMC or sbRIO-9651 Socketed Component Level IP Declaration** drop-down list to select the CLIP you created.
6. On the Clock Selections page, configure any clocks that you specified on the Clock Settings page of the sbRIO CLIP Generator.
7. Click **OK** to close the Component-Level IP Properties dialog box.

◀ **Review:**		**Continue:** ▶
Creating Component-Level IP for the sbRIO-9651		Implementing FPGA VIs for the sbRIO-9651

Related Topics

Creating FPGA VIs (FPGA Module)

Compiling, Downloading, and Running FPGA VIs (FPGA Module)

General Page (sbRIO-9651 Socket Properties Dialog Box)

In the Component-Level IP Properties dialog box for the sbRIO-9651, select **General** from the **Category** list to display this page, which you can use to select a socketed CLIP you created for the sbRIO-9651.

This page includes the following options:

- **sbRIO-9651 Socketed Component Level IP Declaration**—The socketed CLIP to add to the LabVIEW project. The drop-down list includes all

CLIP declarations saved in the following directories, or you can browse for a CLIP declaration saved in a different location:

- `Documents\LabVIEW Data\CompactRIO\CLIPs\sbRIO-9651`—The directory in which the sbRIO CLIP Generator saves created CLIPs.
- `<NI>\Shared\CompactRIO\CLIPs`—The directory in which CLIPs may be distributed by an installer.

Note Leave `<Select CLIP Declaration>` selected if you do not want the socket to use a CLIP. All I/O on the sbRIO-9651 connector will be unused.

Note If the selected CLIP declaration is invalid, you cannot navigate to other pages of or click the **OK** button in the Component-Level IP Properties dialog box. Cancel the Component-Level IP Properties dialog box and correct any errors in the CLIP declaration before you try selecting the CLIP declaration in the Component-Level IP Properties dialog box again.

- **Description**—The description you provided in the sbRIO CLIP Generator for the selected CLIP declaration.

Related Topics

Getting Started with the sbRIO-9651 in LabVIEW

Implementing FPGA VIs for the sbRIO-9651

Complete the following steps to implement a LabVIEW FPGA VI for the sbRIO-9651.

Before You Begin

- Use the sbRIO CLIP Generator to generate a unique socketed CLIP.
- Add the CLIP to a LabVIEW project.
- Connect the sbRIO-9651 to your computer with a USB cable or through a network.

- Configure the sbRIO-9651 with Measurement & Automation Explorer (MAX).

1. Create and compile your FPGA VI.
2. If you enabled any peripherals in your CLIP, such as secondary ethernet, CAN, or serial, download your FPGA VI to the flash of the sbRIO-9651 to load on boot in order to ensure that the driver for the peripheral can load properly at boot time.

You should now be able to run and deploy the system just like any other sbRIO device.

◀ **Review:** Adding a CLIP for the sbRIO-9651 to a LabVIEW Project		

Related Topics

Creating FPGA VIs (FPGA Module)

Compiling, Downloading, and Running FPGA VIs (FPGA Module)

sbRIO-9683

RIO Mezzanine Card, General Purpose Inverter Controller

Software Reference (?)

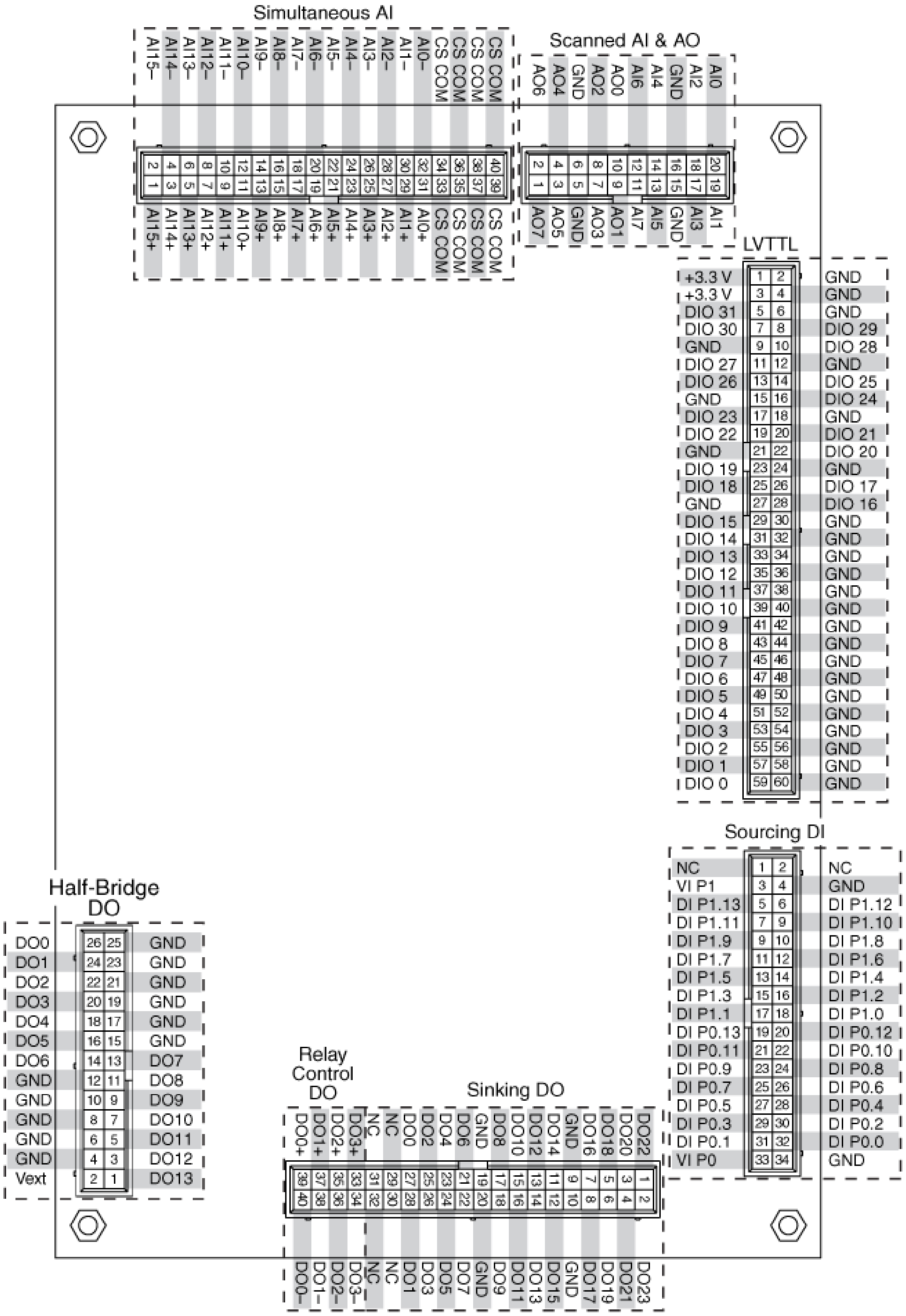
FPGA Interface

Hardware Documentation (?)

sbRIO-9683 hardware documentation on ni.com/manuals

Note Use the sbRIO-9683 with only sbRIO-9605/9606/9607 devices.

sbRIO-9683 Pinout


Simultaneous AI
AI15– AI14– AI13– AI12– AI11– AI10– AI9– AI8– AI7– AI6– AI5– AI4– AI3– AI2– AI1– AI0– CS COM CS COM CS COM CS COM
AI15+ AI14+ AI13+ AI12+ AI11+ AI10+ AI9+ AI8+ AI7+ AI6+ AI5+ AI4+ AI3+ AI2+ AI1+ AI0+ CS COM CS COM CS COM CS COM
Scanned AI & AO
AO6 AO4 GND AO2 AO0 AI6 AI4 GND AI2 AI0
AO7 AO5 GND AO3 AO1 AI7 AI5 GND AI3 AI1
LVTTL
+3.3 V, +3.3 V, DIO 31, DIO 30, GND, DIO 27, DIO 26, GND, DIO 23, DIO 22, GND, DIO 19, DIO 18, GND, DIO 15, DIO 14, DIO 13, DIO 12, DIO 11, DIO 10, DIO 9, DIO 8, DIO 7, DIO 6, DIO 5, DIO 4, DIO 3, DIO 2, DIO 1, DIO 0
GND, GND, GND, DIO 29, DIO 28, GND, DIO 25, DIO 24, GND, DIO 21, DIO 20, GND, DIO 17, DIO 16, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND, GND
Sourcing DI
NC, VI P1, DI P1.13, DI P1.11, DI P1.9, DI P1.7, DI P1.5, DI P1.3, DI P1.1, DI P0.13, DI P0.11, DI P0.9, DI P0.7, DI P0.5, DI P0.3, DI P0.1, VI P0
NC, GND, DI P1.12, DI P1.10, DI P1.8, DI P1.6, DI P1.4, DI P1.2, DI P1.0, DI P0.12, DI P0.10, DI P0.8, DI P0.6, DI P0.4, DI P0.2, DI P0.0, GND
Half-Bridge DO
DO0, DO1, DO2, DO3, DO4, DO5, DO6, GND, GND, GND, GND, GND, Vext
GND, GND, GND, GND, GND, GND, DO7, DO8, DO9, DO10, DO11, DO12, DO13
Relay Control DO
Sinking DO
DO0+ DO1+ DO2+ DO3+ NC NC DO0 DO2 DO4 DO6 GND DO8 DO10 DO12 DO14 GND DO16 DO18 DO20 DO22
DO0– DO1– DO2– DO3– NC NC DO1 DO3 DO5 DO7 GND DO9 DO11 DO13 DO15 GND DO17 DO19 DO21 DO23

sbRIO-9683 (FPGA Interface)

RIO Mezzanine Card

General Purpose Inverter Controller

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
AO**x**	Analog output channel where **x** is the number of the channel. The sbRIO-9683 has AO channels 0 to 7 that you can add to a project.
Half-Bridge DO**x**	Half-bridge digital output channel where **x** is the number of the channel. The sbRIO-9683 has Half-bridge DO channels 0 to 13 that you can add to a project.
Half-Bridge DO13:0	Digital port consisting of Half-bridge DO channels 0 through 13. Channel 13 is returned in the MSB, and channel 0 is returned in the LSB.
Half-Bridge DO7:0	Digital port consisting of Half-bridge DO channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Half-Bridge DO13:8	Digital port consisting of Half-bridge DO channels 8 through 13. Channel 13 is returned in the MSB, and channel 8 is returned in the LSB.
LVTTL DIO**x**	LVTTL digital input/output channel where **x** is the number of the channel. The sbRIO-9683 has LVTTL DIO channels 0 to 31 that you can add to a project.
LVTTL DIO7:0	Digital port consisting of LVTTL DIO channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.

LVTTL DIO15:8	Digital port consisting of LVTTL DIO channels 8 through 15. Channel 15 is returned in the MSB, and channel 8 is returned in the LSB.
LVTTL DIO23:16	Digital port consisting of LVTTL DIO channels 16 through 23. Channel 23 is returned in the MSB, and channel 16 is returned in the LSB.
LVTTL DIO31:24	Digital port consisting of LVTTL DIO channels 24 through 31. Channel 31 is returned in the MSB, and channel 24 is returned in the LSB.
Relay Control DO**x**	Relay control digital output channel where **x** is the number of the channel. The sbRIO-9683 has relay control DO channels 0 to 3 that you can add to a project.
Relay Control DO3:0	Digital port consisting of relay control DO channels 0 through 3. Channel 3 is returned in the MSB, and channel 0 is returned in the LSB.
Scanned AI**x**	Scanned analog input channel where **x** is the number of the channel. The sbRIO-9683 has scanned AI channels 0 to 7 that you can add to a project.
Simultaneous AI**x**	Simultaneous analog input channel where **x** is the number of the channel. The sbRIO-9683 has simultaneous AI channels 0 to 15 that you can add to a project.
Sinking DO**x**	Sinking digital output channel where **x** is the number of the channel. The sbRIO-9683 has sinking DO channels 0 to 23 that you can add to a project.
Sinking DO23:0	Digital port consisting of sinking DO channels 0 through 23. Channel 23 is returned in the MSB, and channel 0 is returned in the LSB.
Sinking DO7:0	Digital port consisting of sinking DO channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Sinking DO15:8	Digital port consisting of sinking DO channels 8 through 15. Channel 15 is returned in the MSB, and channel 8 is returned in the LSB.

Sinking DO23:16	Digital port consisting of sinking DO channels 16 through 23. Channel 23 is returned in the MSB, and channel 16 is returned in the LSB.
Sourcing P0 DI**x**	Sourcing digital input channel where **x** is the number of the channel. The sbRIO-9683 has sourcing P0 DI channels 0 to 13 that you can add to a project.
Sourcing P0 DI13:0	Digital port consisting of sourcing DI P0 channels 0 through 13. Channel 13 is returned in the MSB, and channel 0 is returned in the LSB.
Sourcing P0 DI7:0	Digital port consisting of sourcing DI P0 channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Sourcing P0 DI13:8	Digital port consisting of sourcing DI P0channels 8 through 13. Channel 13 is returned in the MSB, and channel 8 is returned in the LSB.
Sourcing P1 DI**x**	Sourcing digital input channel where **x** is the number of the channel. The sbRIO-9683 has sourcing P1 DI channels 0 to 13 that you can add to a project.
Sourcing P1 DI13:0	Digital port consisting of sourcing DI P1 channels 0 through 13. Channel 13 is returned in the MSB, and channel 0 is returned in the LSB.
Sourcing P1 DI7:0	Digital port consisting of sourcing DI P1 channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Sourcing P1 DI13:8	Digital port consisting of sourcing DI P1 channels 8 through 13. Channel 13 is returned in the MSB, and channel 8 is returned in the LSB.

Arbitration

You can configure the arbitration settings for the LVTTL DIO, half-bridge DO, relay control DO, and sinking DO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate for the LVTTL DIO, half-bridge DO, and relay control DO channels and Arbitrate if Multiple Requestors Only for the sinking DO channels.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for LVTTL DIO channels of this device.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node (FPGA Module) topic for a description of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a description of this method.
Wait on Any Edge	Pauses the execution of the I/O Method Node until the next falling or rising edge of the digital signal. The **Timeout** input specifies in FPGA clock ticks how long the Wait on Any Edge method waits for the next falling or rising edge. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on Falling Edge	Pauses the execution of the I/O Method Node until the next falling edge of the digital signal. The **Timeout** input specifies in FPGA clock ticks how long the Wait on Falling Edge method waits for the next falling edge. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on High Level	Pauses the execution of the I/O Method Node until the digital signal is high. The **Timeout** input specifies in FPGA clock ticks how long the Wait on High Level method waits for the next high level. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on Low Level	Pauses the execution of the I/O Method Node until the digital signal is low. The **Timeout** input s

	pecifies in FPGA clock ticks how long the Wait on Low Level method waits for the next low level. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on Rising Edge	Pauses the execution of the I/O Method Node until the next rising edge of the digital signal. The **Timeout** input specifies in FPGA clock ticks how long the Wait on Rising Edge method waits for the next rising edge. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.

RMC Methods

Use the FPGA I/O Method Node to access the following RMC methods for this device.

Method	Description
Check Status	Returns a Boolean value that indicates whether the half-bridge DO on the device is ready.

I/O Properties

Use the FPGA I/O Property Node to access the following I/O properties for this device.

Property	Description
Voltage Range	Returns the input range for a simultaneous analog input bank.

RMC Properties

Use the FPGA I/O Property Node to access the following RMC properties for this device.

Property	Description

Voltage Range Simultaneous AI**x:x**	Sets the input range for a simultaneous analog input bank as either ±5 V or ±10 V where **x**:x is the bank. This method overwrites the value you configure in the RIO Mezzanine Card Properties dialog box.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop. Configure the number of output synchronizing registers for the high-speed DO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box.

Avoiding Timing Uncertainty with the NI 9683 (FPGA Interface)

Follow these guidelines to avoid introducing timing uncertainty into scanned AI Node and AO Node method calls.

- Use a single I/O Node to access scanned AI and AO operations to ensure proper sequencing.
- Do not perform the following operations concurrently:

– Scanned AI Node call

– AO Node call

Note Simultaneous AI on the NI 9683 does not affect scanned AI and AO and is not affected by scanned AI and AO.

RIO Mezzanine Card Properties Dialog Box for the NI 9683 (FPGA Interface)

Right-click an NI 9683 RIO Mezzanine Card (RMC) in the **Project Explorer** window and select **Properties** from the shortcut menu to display this dialog box.

Use this dialog box to configure an RMC.

This dialog box includes the following components:

- **Name**—Specifies the name of the RMC, which appears in the **Project Explorer** window. LabVIEW assigns a default name to the device. You can use this field to give the device a descriptive name.

- **Type**—Specifies the type of RMC. You cannot change this option.
- **Scaling Mode**—Sets the scaling mode for the RMC. Select **Scaled** if you want the FPGA I/O Node to return scaled, fixed-point data from the device in units of volts. The fixed-point data is signed, with a word length of 20 bits and an integer word length of 5 bits for simultaneous AI and unsigned, with a word length of 16 bits and an integer word length of 3 bits for scanned AI and AO. Select **Raw** if you want the FPGA I/O Node to return unscaled, binary data from the device. If you select **Raw**, you must scale the analog input values in the host VI. The default is **Scaled**.
- **Simultaneous AI - Default Voltage Range**—Specifies the input range for each simultaneous analog input bank. The NI 9683 contains eight banks consisting of two channels each.
- **Sinking DO - Disable Arbitration**—Place a checkmark in this checkbox if you want to change the arbitration setting to Never Arbitrate and reduce the amount of FPGA logic used by VIs. Check this box only if you are sure that the design of the FPGA VI will never allow more than one digital function to execute at the same time, even on different channels. If more than one digital function could execute simultaneously in the FPGA VI, leave the box unchecked to keep the default Arbitrate if Multiple Requestors Only arbitration setting.
- **LVTTL DIO - Initial Line Direction**—Sets the initial line direction for each LVTTL DIO channel to input or output. The default is input.

Scaling NI 9683 Analog Input Values (FPGA Interface)

Set the **Scaling Mode** to **Scaled** in the **RIO Mezzanine Card Properties** dialog box for the NI 9683 if you want the FPGA I/O Node to return scaled, fixed-point data for the device. If you set the **Scaling Mode** to **Raw**, the FPGA I/O Node returns unscaled, binary values for the NI 9683 that you can convert into volts using the following equations. You must convert these values in the host VI

Scanned AI

Use the following equation to convert binary values into volts for the scanned AI on the NI 9683.

Volts = **Binary Value** × 5 V ÷ 4096

where **Binary Value** is the signed or unsigned value returned by the FPGA I/O Node

Simultaneous AI

Use the following equations to convert binary values into volts for the simultaneous AI on the NI 9683.

±10 V Range

Volts = **Binary Value** × 20 V ÷ 4096

where **Binary Value** is the signed or unsigned value returned by the FPGA I/O Node

±5 V Range

Volts = **Binary Value** × 10 V ÷ 4096

where **Binary Value** is the signed or unsigned value returned by the FPGA I/O Node

Scaling NI 9683 Analog Output Values (FPGA Interface)

Set the **Scaling Mode** to **Scaled** in the **RIO Mezzanine Card Properties** dialog box for the NI 9683 if you want the FPGA I/O Node to accept fixed-point data in volts when writing to the device. If you set the **Scaling Mode** to **Raw**, the FPGA I/O Node accepts only binary values. You can convert output voltage values to binary values using the following equation. You must convert these values in the host VI.

Binary Value = **Volts** × 4096 ÷ 5 V

where **Binary Value** is the value you write to the FPGA I/O Node

sbRIO-9684

RIO Mezzanine Card, 16-bit General Purpose Inverter Controller

Software Reference (?)

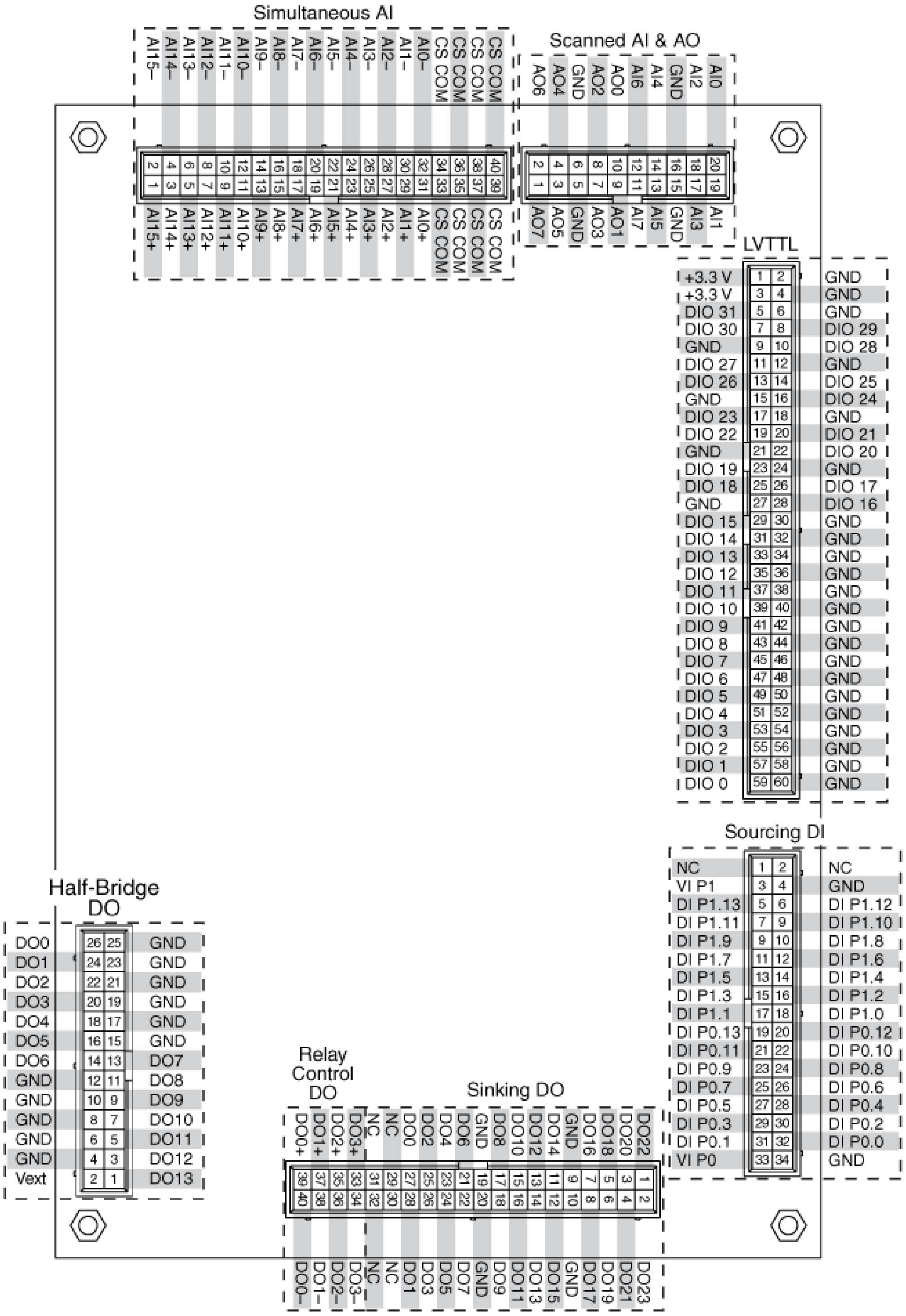
FPGA Interface

Hardware Documentation (?)

sbRIO-9684 hardware documentation on ni.com/manuals

Note Use the sbRIO-9684 with only the sbRIO-9607.

sbRIO-9684 Pinout



sbRIO-9684 (FPGA Interface)

RIO Mezzanine Card

16-bit General Purpose Inverter Controller

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
AO**x**	Analog output channel where **x** is the number of the channel. The sbRIO-9684 has AO channels 0 to 7 that you can add to a project.
Half-Bridge DO**x**	Half-bridge digital output channel where **x** is the number of the channel. The sbRIO-9684 has Half-bridge DO channels 0 to 13 that you can add to a project.
Half-Bridge DO13:0	Digital port consisting of Half-bridge DO channels 0 through 13. Channel 13 is returned in the MSB, and channel 0 is returned in the LSB.
Half-Bridge DO7:0	Digital port consisting of Half-bridge DO channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Half-Bridge DO13:8	Digital port consisting of Half-bridge DO channels 8 through 13. Channel 13 is returned in the MSB, and channel 8 is returned in the LSB.
LVTTL DIO**x**	LVTTL digital input/output channel where **x** is the number of the channel. The sbRIO-9684 has LVTTL DIO channels 0 to 31 that you can add to a project.
LVTTL DIO7:0	Digital port consisting of LVTTL DIO channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.

LVTTL DIO15:8	Digital port consisting of LVTTL DIO channels 8 through 15. Channel 15 is returned in the MSB, and channel 8 is returned in the LSB.
LVTTL DIO23:16	Digital port consisting of LVTTL DIO channels 16 through 23. Channel 23 is returned in the MSB, and channel 16 is returned in the LSB.
LVTTL DIO31:24	Digital port consisting of LVTTL DIO channels 24 through 31. Channel 31 is returned in the MSB, and channel 24 is returned in the LSB.
Relay Control DO**x**	Relay control digital output channel where **x** is the number of the channel. The sbRIO-9684 has relay control DO channels 0 to 3 that you can add to a project.
Relay Control DO3:0	Digital port consisting of relay control DO channels 0 through 3. Channel 3 is returned in the MSB, and channel 0 is returned in the LSB.
Scanned AI**x**	Scanned analog input channel where **x** is the number of the channel. The sbRIO-9684 has scanned AI channels 0 to 7 that you can add to a project.
Simultaneous AI**x**	Simultaneous analog input channel where **x** is the number of the channel. The sbRIO-9684 has simultaneous AI channels 0 to 15 that you can add to a project.
Sinking DO**x**	Sinking digital output channel where **x** is the number of the channel. The sbRIO-9684 has sinking DO channels 0 to 23 that you can add to a project.
Sinking DO23:0	Digital port consisting of sinking DO channels 0 through 23. Channel 23 is returned in the MSB, and channel 0 is returned in the LSB.
Sinking DO7:0	Digital port consisting of sinking DO channels 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Sinking DO15:8	Digital port consisting of sinking DO channels 8 through 15. Channel 15 is returned in the MSB, and channel 8 is returned in the LSB.

Sinking DO23:16	Digital port consisting of sinking DO channels 16 through 23. Channel 23 is returned in the MSB, a nd channel 16 is returned in the LSB.
Sourcing P0 DI**x**	Sourcing digital input channel where **x** is the nu mber of the channel. The sbRIO-9684 has sourci ng P0 DI channels 0 to 13 that you can add to a p roject.
Sourcing P0 DI13:0	Digital port consisting of sourcing DI P0 channel s 0 through 13. Channel 13 is returned in the MS B, and channel 0 is returned in the LSB.
Sourcing P0 DI7:0	Digital port consisting of sourcing DI P0 channel s 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Sourcing P0 DI13:8	Digital port consisting of sourcing DI P0channels 8 through 13. Channel 13 is returned in the MSB, and channel 8 is returned in the LSB.
Sourcing P1 DI**x**	Sourcing digital input channel where **x** is the nu mber of the channel. The sbRIO-9684 has sourci ng P1 DI channels 0 to 13 that you can add to a p roject.
Sourcing P1 DI13:0	Digital port consisting of sourcing DI P1 channel s 0 through 13. Channel 13 is returned in the MS B, and channel 0 is returned in the LSB.
Sourcing P1 DI7:0	Digital port consisting of sourcing DI P1 channel s 0 through 7. Channel 7 is returned in the MSB, and channel 0 is returned in the LSB.
Sourcing P1 DI13:8	Digital port consisting of sourcing DI P1 channel s 8 through 13. Channel 13 is returned in the MS B, and channel 8 is returned in the LSB.

Arbitration

You can configure the arbitration settings for the LVTTL DIO, half-bridge DO, relay control DO, and sinking DO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate for the LVTTL DIO, half-bridge DO, and relay control DO channels and Arbitrate if Multiple Requestors Only for the sinking DO channels.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for LVTTL DIO channels of this device.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node (FPGA Module) topic for a description of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a description of this method.
Wait on Any Edge	Pauses the execution of the I/O Method Node until the next falling or rising edge of the digital signal. The **Timeout** input specifies in FPGA clock ticks how long the Wait on Any Edge method waits for the next falling or rising edge. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on Falling Edge	Pauses the execution of the I/O Method Node until the next falling edge of the digital signal. The **Timeout** input specifies in FPGA clock ticks how long the Wait on Falling Edge method waits for the next falling edge. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on High Level	Pauses the execution of the I/O Method Node until the digital signal is high. The **Timeout** input specifies in FPGA clock ticks how long the Wait on High Level method waits for the next high level. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on Low Level	Pauses the execution of the I/O Method Node until the digital signal is low. The **Timeout** input s

	pecifies in FPGA clock ticks how long the Wait on Low Level method waits for the next low level. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.
Wait on Rising Edge	Pauses the execution of the I/O Method Node until the next rising edge of the digital signal. The **Timeout** input specifies in FPGA clock ticks how long the Wait on Rising Edge method waits for the next rising edge. A value of 0 causes the method to time out immediately, a negative value causes the method to wait indefinitely, and a positive value causes the method to wait for that number of clock ticks before timing out.

RMC Methods

Use the FPGA I/O Method Node to access the following RMC methods for this device.

Method	Description
Check Status	Returns a Boolean value that indicates whether the half-bridge DO on the device is ready.

I/O Properties

Use the FPGA I/O Property Node to access the following I/O properties for this device.

Property	Description
LSB Weight (±10V range)	Returns the LSB weight in V/LSB for the ±10V range.
LSB Weight (±5V range)	Returns the LSB weight in V/LSB for the ±5V range.
Offset (±10V range)	Returns the calibration offset in V for the ±10V range.
Offset (±5V range)	Returns the calibration offset in V for the ±5V range.

Voltage Range	Returns the input range for a simultaneous analog input bank.

RMC Properties

Use the FPGA I/O Property Node to access the following RMC properties for this device.

Property	Description
Voltage Range Simultaneous AI**x**:**x**	Sets the input range for a simultaneous analog input bank as either ±5 V or ±10 V where **x**:x is the bank. This method overwrites the value you configure in the RIO Mezzanine Card Properties dialog box.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop. Configure the number of output synchronizing registers for the high-speed DO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box.

Avoiding Timing Uncertainty with the sbRIO-9684 (FPGA Interface)

Follow these guidelines to avoid introducing timing uncertainty into scanned AI Node and AO Node method calls.

- Use a single I/O Node to access scanned AI and AO operations to ensure proper sequencing.
- Do not perform the following operations concurrently:

– Scanned AI Node call

– AO Node call

Note Simultaneous AI on the sbRIO-9684 does not affect scanned AI and AO and is not affected by scanned AI and AO.

RIO Mezzanine Card Properties Dialog Box for the sbRIO-9684 (FPGA Interface)

Right-click an sbRIO-9684 RIO Mezzanine Card (RMC) in the **Project Explorer** window and select **Properties** from the shortcut menu to display this dialog box.

Use this dialog box to configure an RMC.

This dialog box includes the following components:

- **Name**—Specifies the name of the RMC, which appears in the **Project Explorer** window. LabVIEW assigns a default name to the device. You can use this field to give the device a descriptive name.
- **Type**—Specifies the type of RMC. You cannot change this option.
- **Scaling Mode**—Sets the scaling mode for the RMC. Select **Scaled** if you want the FPGA I/O Node to return scaled, fixed-point data from the device in units of volts. The fixed-point data is signed, with a word length of 20 bits and an integer word length of 5 bits for simultaneous AI and unsigned, with a word length of 16 bits and an integer word length of 3 bits for scanned AI and AO. Select **Raw** if you want the FPGA I/O Node to return unscaled, binary data from the device. If you select **Raw**, you must scale the analog input values in the host VI. The default is **Scaled**.
- **Simultaneous AI - Default Voltage Range**—Specifies the input range for each simultaneous analog input bank. The sbRIO-9684 contains eight banks consisting of two channels each.
- **Sinking DO - Disable Arbitration**—Place a checkmark in this checkbox if you want to change the arbitration setting to Never Arbitrate and reduce the amount of FPGA logic used by VIs. Check this box only if you are sure that the design of the FPGA VI will never allow more than one digital function to execute at the same time, even on different channels. If more than one digital function could execute simultaneously in the FPGA VI, leave the box unchecked to keep the default Arbitrate if Multiple Requestors Only arbitration setting.
- **LVTTL DIO - Initial Line Direction**—Sets the initial line direction for each LVTTL DIO channel to input or output. The default is input.

Scaling sbRIO-9684 Analog Input Values (FPGA Interface)

Set the **Scaling Mode** to **Scaled** in the **RIO Mezzanine Card Properties** dialog box for the NI 9684 if you want the FPGA I/O Node to return scaled, fixed-point data for the device. If you set the **Scaling Mode** to **Raw**, the FPGA I/O Node returns unscaled, binary values for the NI 9684 that you can convert into volts using the following equations. You must convert these values in the host VI

Scanned AI

Use the following equation to convert binary values into volts for the scanned AI on the sbRIO-9684.

Volts = **Binary Value** × 5 V ÷ 4096

where

Binary Value is the signed or unsigned value returned by the FPGA I/O Node

Simultaneous AI

Use the following equations to convert binary values into volts for the simultaneous AI on the sbRIO-9684 for both ±10V and ±5V ranges.

Volts = (**Binary Value** × LSB Weight + Offset)*

where

Binary Value is the signed value returned by the FPGA I/O Node

LSB Weight is the value returned by the LSB Weight property

Offset is the value returned by the Offset property

If you do not want to read the LSB Weight and Offset values from the module, you can convert to uncalibrated engineering units by using the following values for Offset and LSB Weight:

Offset = 0

LSB Weight = **Typical Input Span** ÷ $2^{\textbf{ADC Resolution}}$

where

Typical Input Span is 20.832V for ±10V range and 10.416V for ±5V range

ADC Resolution is the ADC resolution value in the hardware documentation for the module

*LSB Weight and Offset are formatted and cannot be used as is. Refer to the Binary to Nominal VI in the `LabVIEW\examples\CompactRIO\RMC Specific\NI 9684\NI 9684 High Speed Acquisition\NI 9684 High Speed Acquisition\NI 9684 High Speed Acquisition.lvproj` for an example on how to convert the LSB Weight and Offset to the correct format.

Scaling sbRIO-9684 Analog Output Values (FPGA Interface)

Set the **Scaling Mode** to **Scaled** in the **RIO Mezzanine Card Properties** dialog box for the NI 9684 if you want the FPGA I/O Node to accept fixed-point data in volts when writing to the device. If you set the **Scaling Mode** to **Raw**, the FPGA I/O Node accepts only binary values. You can convert output voltage values to binary values using the following equation. You must convert these values in the host VI.

Binary Value = **Volts** × 4096 ÷ 5 V

where

Binary Value is the value you write to the FPGA I/O Node

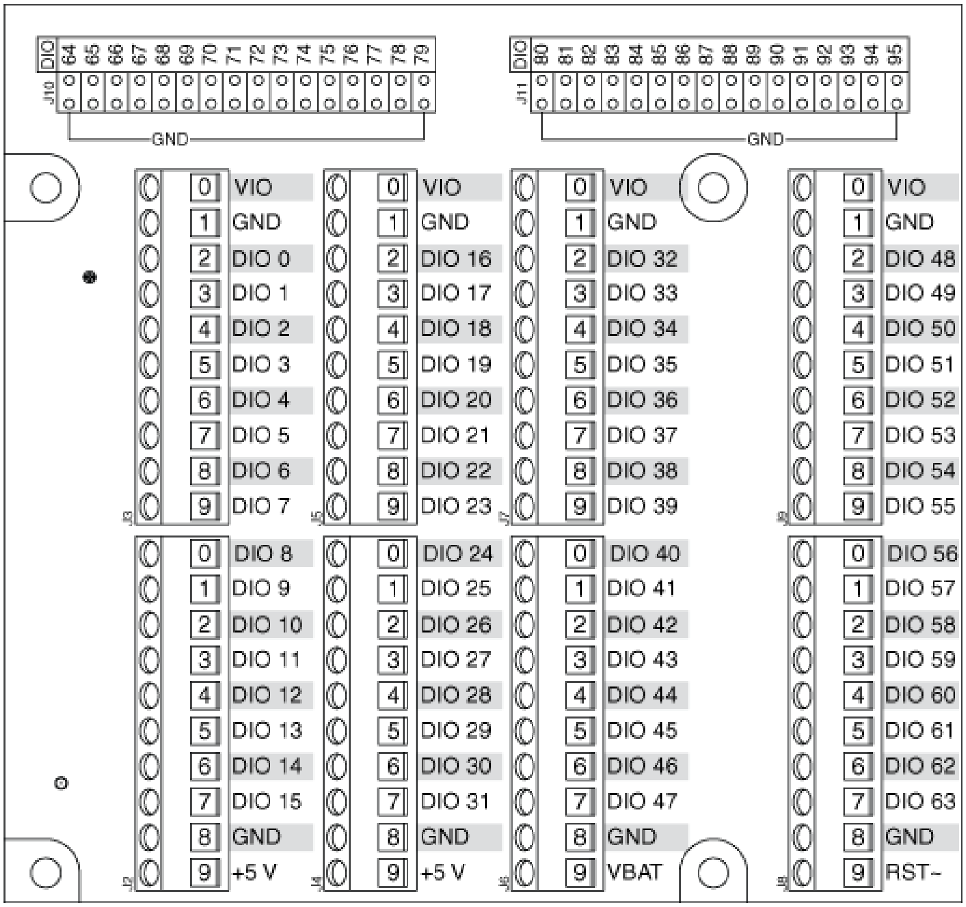
sbRIO-9694

RIO Mezzanine Card, 96 channel digital I/O breakout board

Software Reference (?)

FPGA Interface

sbRIO-9694 Pinout



sbRIO-9694 (FPGA Interface)

RIO Mezzanine Card

96 channel digital I/O breakout board

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description

DIO**x**	Digital input/output channel where **x** is the number of the channel. The sbRIO-9694 has DIO channels 0 to 95 that you can add to a project.
DIO15:0	Digital port consisting of DIO channels 0 through 15. Channel 15 is returned in the MSB, and channel 0 is returned in the LSB.
DIO31:16	Digital port consisting of DIO channels 16 through 31. Channel 31 is returned in the MSB, and channel 16 is returned in the LSB.
DIO47:32	Digital port consisting of DIO channels 32 through 47. Channel 47 is returned in the MSB, and channel 32 is returned in the LSB.
DIO63:48	Digital port consisting of DIO channels 48 through 63. Channel 63 is returned in the MSB, and channel 48 is returned in the LSB.
DIO79:64	Digital port consisting of DIO channels 64 through 79. Channel 79 is returned in the MSB, and channel 64 is returned in the LSB.
DIO95:80	Digital port consisting of DIO channels 80 through 95. Channel 95 is returned in the MSB, and channel 80 is returned in the LSB.

Arbitration

You can configure the arbitration settings for the DIO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

This device does not support any module methods.

Properties

This device does not support any properties.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop. Configure the number of output synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. Configure the number of input synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Node Properties dialog box.

sbRIO-9697

RIO Mezzanine Card

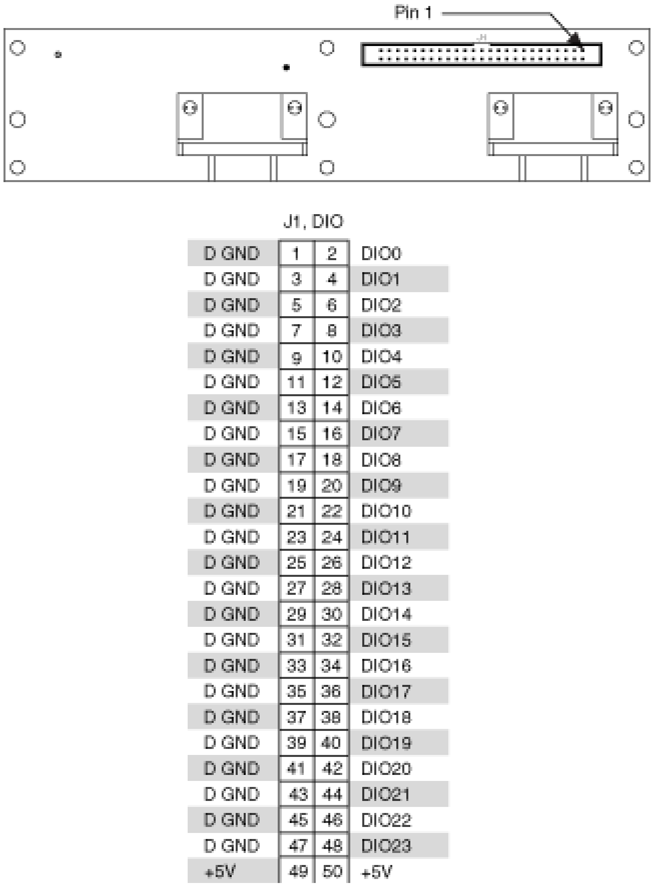
2-slot C Series, 24 3.3 V DIO channels

Software Reference (?)

FPGA Interface

Note Use the sbRIO-9697 with only sbRIO-9607/9627 devices.

sbRIO-9697 Pinout



J1, DIO

D GND	1	2	DIO0
D GND	3	4	DIO1
D GND	5	6	DIO2
D GND	7	8	DIO3
D GND	9	10	DIO4
D GND	11	12	DIO5
D GND	13	14	DIO6
D GND	15	16	DIO7
D GND	17	18	DIO8
D GND	19	20	DIO9
D GND	21	22	DIO10
D GND	23	24	DIO11
D GND	25	26	DIO12
D GND	27	28	DIO13
D GND	29	30	DIO14
D GND	31	32	DIO15
D GND	33	34	DIO16
D GND	35	36	DIO17
D GND	37	38	DIO18
D GND	39	40	DIO19
D GND	41	42	DIO20
D GND	43	44	DIO21
D GND	45	46	DIO22
D GND	47	48	DIO23
+5V	49	50	+5V

sbRIO-9697 (FPGA Interface)

2-slot C Series RIO Mezzanine Card

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description

DIO**x**	Digital input/output channel where **x** is the number of the channel. The sbRIO-9697 has DIO channels 0 to 23 that you can add to a project.
DIO15:0	Digital port consisting of DIO channels 0 through 15. Channel 15 is returned in the MSB, and channel 0 is returned in the LSB.
DIO23:16	Digital port consisting of DIO channels 16 through 23. Channel 23 is returned in the MSB, and channel 16 is returned in the LSB.

Arbitration

You can configure the arbitration settings for the DIO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

This device does not support any module methods.

Properties

This device does not support any properties.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop. Configure the number of output synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. Configure the number of input synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Node Properties dialog box.

sbRIO-9698

RIO Mezzanine Card

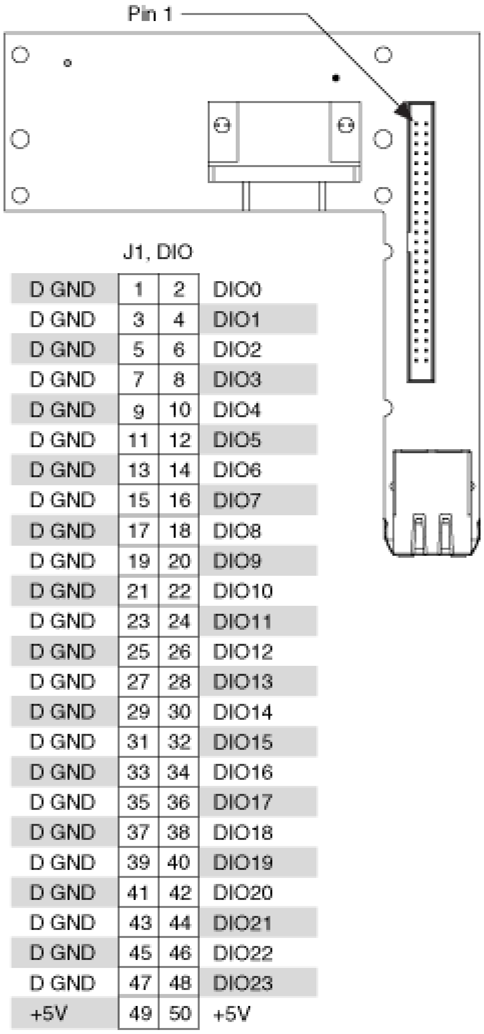
1-Slot C Series, Secondary Ethernet Port, 24 3.3 V DIO channels

Software Reference (?)

FPGA Interface

Note Use the sbRIO-9698 with only sbRIO-9607/9627 devices.

sbRIO-9698 Pinout



D GND	1	2	DIO0
D GND	3	4	DIO1
D GND	5	6	DIO2
D GND	7	8	DIO3
D GND	9	10	DIO4
D GND	11	12	DIO5
D GND	13	14	DIO6
D GND	15	16	DIO7
D GND	17	18	DIO8
D GND	19	20	DIO9
D GND	21	22	DIO10
D GND	23	24	DIO11
D GND	25	26	DIO12
D GND	27	28	DIO13
D GND	29	30	DIO14
D GND	31	32	DIO15
D GND	33	34	DIO16
D GND	35	36	DIO17
D GND	37	38	DIO18
D GND	39	40	DIO19
D GND	41	42	DIO20
D GND	43	44	DIO21
D GND	45	46	DIO22
D GND	47	48	DIO23
+5V	49	50	+5V

sbRIO-9698 (FPGA Interface)

1-Slot C Series, Secondary Ethernet Port, 24 3.3 V DIO channels

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals for this device.

Terminal	Description
DIO**x**	Digital input/output channel where **x** is the number of the channel. The sbRIO-9698 has DIO channels 0 to 23 that you can add to a project.
DIO15:0	Digital port consisting of DIO channels 0 through 15. Channel 15 is returned in the MSB, and channel 0 is returned in the LSB.
DIO23:16	Digital port consisting of DIO channels 16 through 23. Channel 23 is returned in the MSB, and channel 16 is returned in the LSB.

Arbitration

You can configure the arbitration settings for the DIO channels of this device in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods for this device.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

This device does not support any module methods.

Properties

This device does not support any properties.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop. Configure the number of output synchronizing registers for the channels of this device in the Advanced Code

Generation page of the FPGA I/O Properties dialog box. Configure the number of input synchronizing registers for the channels of this device in the Advanced Code Generation page of the FPGA I/O Node Properties dialog box.

NI Digital I/O Mezzanine Card

96 DIO channels

Software Reference (?)

FPGA Interface

NI Digital I/O Mezzanine Card (FPGA Interface)

96 DIO channels

FPGA I/O Node

You can use an FPGA I/O Node, configured for reading and writing, with this device.

Terminals in Software

Use the FPGA I/O Node to access the following terminals.

Terminal	Description
DIO**x**	Digital input/output channel where x is the number of the channel. The digital I/O mezzanine card exposes DIO channels 0 to 95 that you can add to a project.
DIO15:0	Digital port consisting of DIO channels 0 through 15. Channel 15 is returned in the MSB, and channel 0 is returned in the LSB.
DIO31:16	Digital port consisting of DIO channels 16 through 31. Channel 31 is returned in the MSB, and channel 16 is returned in the LSB.
DIO47:32	Digital port consisting of DIO channels 32 through 47. Channel 47 is returned in the MSB, and channel 32 is returned in the LSB.

DIO63:48	Digital port consisting of DIO channels 48 through 63. Channel 63 is returned in the MSB, and channel 48 is returned in the LSB.
DIO79:64	Digital port consisting of DIO channels 64 through 79. Channel 79 is returned in the MSB, and channel 64 is returned in the LSB.
DIO95:80	Digital port consisting of DIO channels 80 through 95. Channel 95 is returned in the MSB, and channel 80 is returned in the LSB.

Arbitration

You can configure the arbitration settings for the DIO channels in the Advanced Code Generation page of the FPGA I/O Properties dialog box. The default arbitration setting is Never Arbitrate.

I/O Methods

Use the FPGA I/O Method Node to access the following I/O methods.

Method	Description
Set Output Data	Refer to the FPGA I/O Method Node topic for a description of this method.
Set Output Enable	Refer to the FPGA I/O Method Node topic for a description of this method.

Module Methods

This device does not support any module methods.

Properties

This device does not support any properties.

Single-Cycle Timed Loop

This device supports the single-cycle Timed Loop. Configure the number of output synchronizing registers for the channels in the Advanced Code Generation page of the FPGA I/O Properties dialog box. Configure the number of input synchronizing registers for the channels in the Advanced Code Generation page of the FPGA I/O Node Properties dialog box.