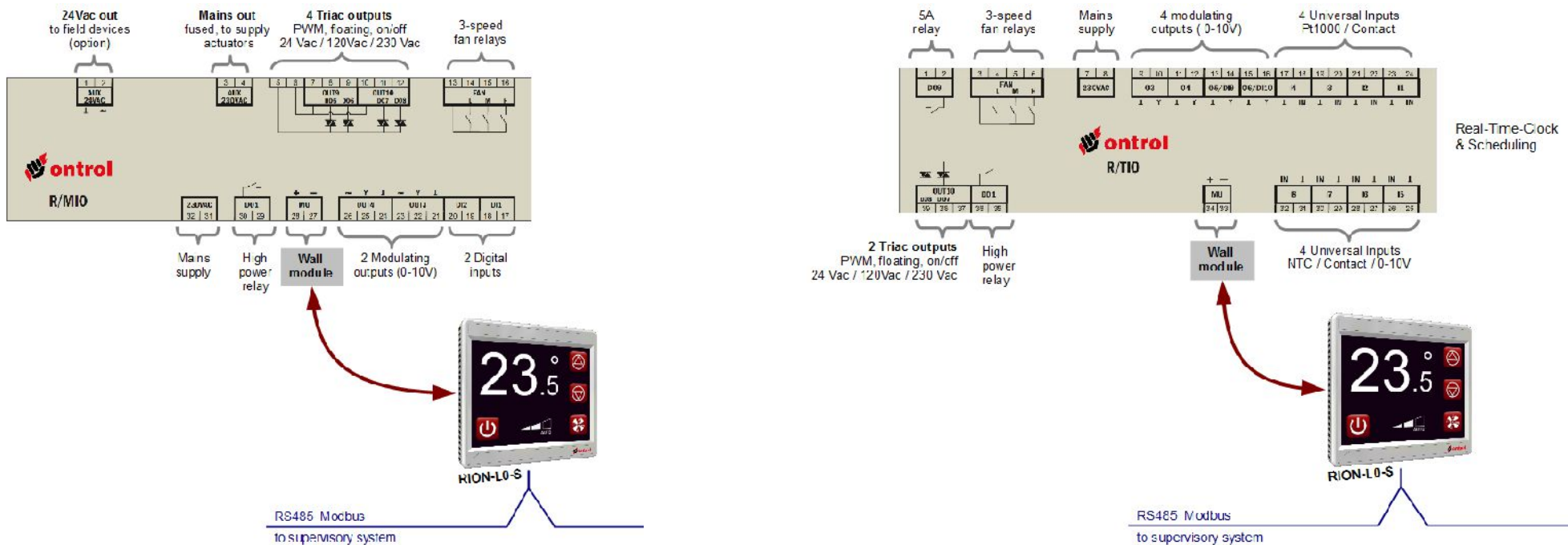




# Using Ontrol dedicated IO devices with the R-ION





# R/MIO and R/TIO Devices



**R/MIO and R/TIO are dedicated input/output modules only for use with the R-ION room controllers.**

## Warning 1:

This is *\*not\** a multi-drop bus, it is a one-to-one connection. Do not connect more than one IO module to the R-ION<sup>1</sup>.

## Warning 2:

These devices are only compatible with the RION-L0-... versions. (See R-ION documentation for details and different versions.)



Two-wire connection for both communications and power



<sup>1</sup> for high IO count applications, you can use different versions of the R-ION with additional modbus ports and standard modbus IO devices.



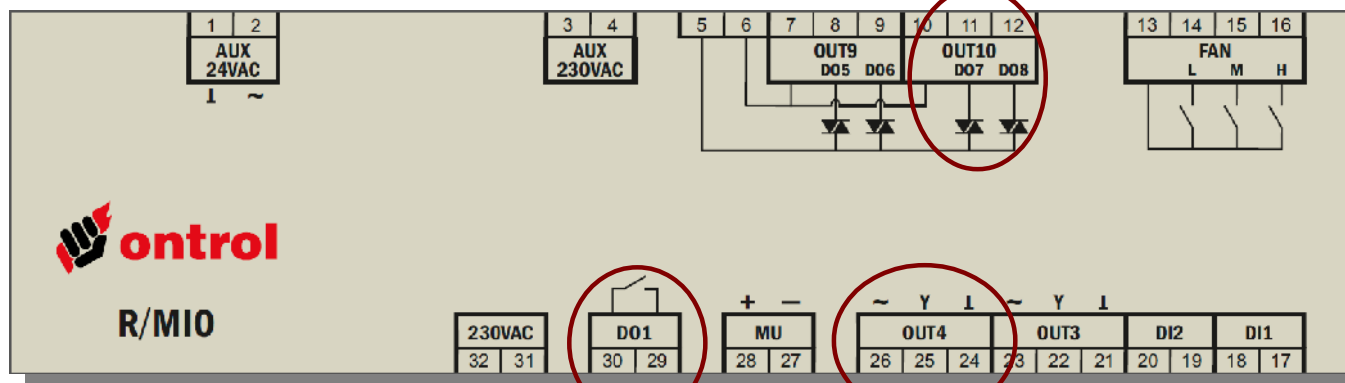
# I/O Channel addresses



I/O channel addresses are printed on the device label, as well as datasheets.

Here are a few examples :

These triacs can be used individually with channel addresses 7 & 8; or as a pair (for floating control) with channel address 10



Relay output at channel address 1

Modulating output at channel address 4

For add-on relay modules, channel addresses are:  
- 11...14 for the R/MIO  
- 11...18 for the R/TIO



# Add 'IoDeviceNetwork' to App



## Add a 'IoDeviceNetwork' component from the ontrolDeviceBus kit to your app.

Set properties as shown on the right:

- Enabled : true
- Serial Port: 250
- Baud Rate: 1200
- Databits : 8
- Stopbit : 1
- Parity : None
- WriteOnStart : true

The screenshot shows the software interface for configuring an 'IoDeviceNetwork' component. The left pane displays a project tree with 'IoNetw' selected under the 'App' folder. The 'ontrolDeviceBus' kit is selected in the component palette, and 'IoDeviceNetwork [184 B]' is highlighted. The right pane shows the configuration properties for 'IoNetw (ontrolDeviceBus::IoDeviceNetwork)'. A red arrow points from the 'IoNetw' component in the tree to the configuration pane.

Property	Value
Meta	Group [1] >>
Ping Time Sec	30 s
Serial Port	250 [0 - 255]
Enabled	<input checked="" type="checkbox"/> true
Ping Enabled	<input checked="" type="checkbox"/> true
Baudrate	1200 [1200 - 57600]
Databits	8 [7 - 8]
Stopbit	1 [1 - 2]
Parity	None
Slow Rate	30 s [30 - 250]
Normal Rate	5 s [5 - 29]
Fast Rate	1 s [0 - 4]
Max Write Time	180 s [0 - 250]
Min Write Time	0 s [0 - 250]
Write On Start	<input checked="" type="checkbox"/> true
Response Timeout	1000 ms [50 - 3000]
Retry Count	1 [0 - 10]



# Add IO Device component to network



**Add a 'RMio' or 'Rtio' component from the ontrolDeviceBus kit to the IoDeviceNetwork.**

Set the slave address to "1"

There should be one - and only one - device under the IoDeviceNetwork.

The screenshot displays a software configuration environment. On the left, a project tree shows a hierarchy: My Network > My File System > My Modules > Platform > Station (bacnet) > Sedona (Fancoil Control):/b > Sedona Tools > App > service > driver > IoNetw > RMio. A red arrow points from the 'RMio' component in the tree to the configuration pane on the right. The configuration pane is titled 'RMio (ontrolDeviceBus::RMio)' and contains several settings: 'Meta' (Group [1]), 'Fault' (false), 'Status' (0), 'Enable' (true), and 'Slave Address' (0 [1 - 247]). Below the configuration pane, a 'Sedona Palette' shows a list of components: AI [68 B], AO [64 B], DI [64 B], DO [72 B], Fan [60 B], FloatingOutput [88 B], IoDeviceNetwork [184 B], PwmOutput [80 B], RMio [80 B] (highlighted), RTio [80 B], and TimeDate [72 B]. The 'ontrolDeviceBus' component is circled in the palette.



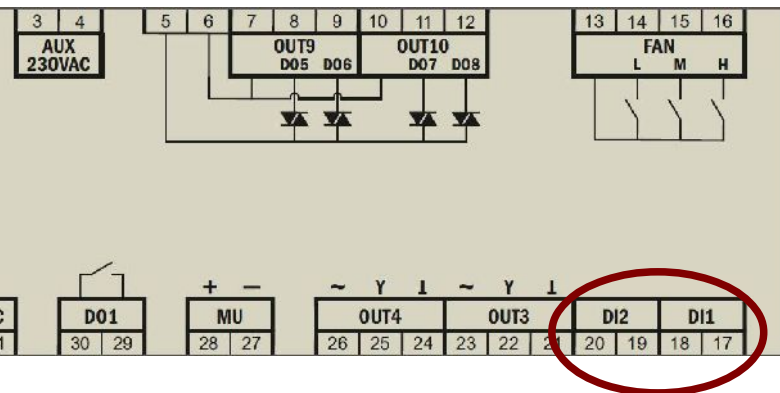


# Reading Digital Inputs



Add a 'DI' component from the ontrolDeviceBus kit to the device.

Set the channel address (see page 3).



Nav

My Network

My File System

My Modules

Platform

Station (bacnet)

Sedona (ExampleFCU):/b

Sedona Tools

App

service

driver

IoNetw

RMio

DI

fancoil

192.168.1.115 (P-ION Default)

Sedona Palette

ontrolDeviceBus

AI [68 B]

AO [64 B]

DI [64 B]

DO [72 B]

Fan [60 B]

FloatingOutput [88 B]

IoDeviceNetwork [184 B]

PwmOutput [80 B]

RMio [80 B]

RTio [80 B]

TimeDate [72 B]

DI (ontrolDeviceBus::DI)

Meta Group [1] >>

Channel No 1 [0 - 255]

Status 0

Out false

In case of an invalid channel number, the 'out' slot will be null.



# Reading Universal Inputs



Add a 'AI' component from the ontrolDeviceBus kit to the device.

Set the channel address (see page3) and input type.

## Input Type selection

- If input type is 'contact' use the ContactOut slot to read the input as a boolean.
- Otherwise, use the AnalogOut slot to read the input as a numeric value.

## Output scaling & unit

- For Pt1000 and NTC input types, output value is in degrees Centigrade.
- For Resistance input types, output value is in ohms.
- For Voltage input types, output value is 0...100%.

The screenshot shows the software interface for configuring an AI component. The left pane displays a project tree with the following structure:

- My File System
- My Modules
- Platform
- Station (bacnet)
- Sedona (ExampleFCU);/b
  - Sedona Tools
  - App
    - service
    - driver
      - IoNetw
      - RTio
      - AI (selected)
    - fancoil

- 192.168.1.115 (P-ION Default)

The right pane shows the configuration for the selected AI component:

- AI (ontrolDeviceBus::AI)
- Meta: Group [1] >>
- Channel No: 1 [0 - 255]
- Status: 131074
- Analog Out: nan
- Contact Out: null
- Input Type: Disabled (dropdown menu is open showing options: Disabled, Contact, PT1000, V010V, V210V, ResistanceLow, ResistanceHigh, NTC10K3)

A red arrow points from the 'ontrolDeviceBus' component in the left pane to the configuration pane on the right.

In case of an invalid channel number, the 'out' slot will be nan or null.



# Controlling Analog Outputs



**Add a 'AO' component from the ontrolDeviceBus kit to the device.**

Set the channel address (see page 3).

Set the output type (0-10V or 2-10V) and the direct/reverse selection.

Link the controlling logic, - in the numeric range 0...100% - to the 'in' slot.

In direct mode:

- 0% input results in 0V (or 2V) output
- 100% input results in 10V output.

In reverse mode:

- 0% input results in 10V output
- 100% input results in 0V (or 2V) output.

The screenshot shows the Sedona software interface. On the left, a tree view displays the project structure, including 'My File System', 'My Modules', 'Platform', 'Station (bacnet)', 'Sedona (ExampleFCU);/b', 'Sedona Tools', 'App', 'service', 'driver', 'IoNetw', 'RTio', 'AO', and 'fancoil'. The 'AO' component is highlighted with a red circle. Below the tree view is the 'Sedona Palette' with a search bar containing 'ontrolDeviceBus'. A red arrow points from the 'AO' component in the tree to the 'AO' component in the palette. The palette lists various components: AI [68 B], AO [64 B], DI [64 B], DO [72 B], Fan [60 B], FloatingOutput [88 B], IoDeviceNetwork [184 B], PwmOutput [80 B], RMio [80 B], RTio [80 B], and TimeDate [72 B]. On the right, the configuration panel for the 'AO (ontrolDeviceBus::AO)' component is shown. It includes the following settings:

- Meta: Group [1] >>
- Channel No: 3 [0 - 255]
- Status: 0
- In: 0 [0 - 100]
- Overrider: nan
- Direct Reverse: Direct
- Output Type: 0-10V (selected), 0-10V, 2-10V, null





# Controlling relays (and individual triacs in on/off mode)



**Add a 'DO' component from the ontrolDeviceBus kit to the device.**

Set the channel address (see page 3).

If the channel address corresponds to a triac output, the triac is driven in on/off mode like a relay.

Link the controlling logic to the 'in' slot to control the relay/triac.

For add-on relay modules,  
channel addresses are:

- 11...14 for the R/MIO
- 11...18 for the R/TIO

The screenshot displays the Sedona software interface. On the left, a tree view shows the project structure under 'My Network'. A red arrow points from the 'DO' component in the 'App' folder to the 'ontrolDeviceBus' component in the 'Sedona Palette'. The 'ontrolDeviceBus' component is highlighted in the palette. On the right, the configuration panel for the 'DO (ontrolDeviceBus::DO)' component is shown. It includes the following settings:

- Meta Group [1] >>
- Channel No: 1
- Status: 0
- In: false
- Overrider: null

Below the configuration panel, a list of components is shown, with 'DO [72 B]' highlighted in blue. Other components include AI [68 B], AO [64 B], DI [64 B], Fan [60 B], FloatingOutput [88 B], IoDeviceNetwork [184 B], PwmOutput [80 B], RMio [80 B], RTio [80 B], and TimeDate [72 B].



# Controlling 3-speed fan



**Add a 'Fan' component from the ontrolDeviceBus kit to the device.**

Link the controlling logic, in the numeric range 0...100%, to the 'inPercentage' slot.

The 3 fan-speed relays are activated as follows:

0%	all off
1...33%	low speed relay
34...66%	medium speed relay
67...100%	high speed relay

The 'out' slot provides an integer type feedback on current fan-speed in the range 0 to 3.

The screenshot displays the configuration of a Fan component. The left pane shows a project tree with the following structure:

- Nav
  - My Network
  - My File System
  - My Modules
  - Platform
  - Station (bacnet)
  - Sedona (ExampleFCU):/b
    - Sedona Tools
    - App
      - service
      - driver
        - IoNetw
        - RTio
        - Fan
      - fancoil
- 192.168.1.115 (P-ION Default)

The right pane shows the configuration for the Fan component:

- Group [1] >>
- In Percentage: 33 [0 - 100]
- Override: nan
- Out: 1 [0 - 255]

The 'ontrolDeviceBus' component is highlighted in the Sedona Palette, and the 'Fan' component is selected in the project tree.



# Controlling individual triacs in PWM mode <sup>1</sup>



Add a 'PwmOutput' component from the ontrolDeviceBus kit to the device.

Set the channel address (see page 3).

Set the PWM period in seconds.

Link the controlling logic, in the numeric range 0...100%, to the 'in' slot.

## PWM operation example

With Pwm period set to 20 seconds and an input value of 25%, the output will be ON for 5 seconds, OFF for 15 seconds.

The screenshot shows the control software interface. On the left, a tree view displays the project structure, including 'Sedona (ExampleFCU):/b' and 'App'. The 'PwmOutp' component is highlighted in the tree. Below the tree, the 'Sedona Palette' shows the 'ontrolDeviceBus' kit selected, with a list of components including 'PwmOutput [80 B]'. On the right, the configuration panel for 'PwmOutp (ontrolDeviceBus::PwmOutput)' is shown, with the following settings:

Property	Value	Range
Channel No	7	[0 - 255]
Status	0	
In	0	[0 - 100]
Pwm Period	20	s [0 - 255]
Override	nan	

<sup>1</sup> commonly used with thermoelectric valve actuators.



# Controlling triac pairs in floating control mode<sup>1</sup>



Add a 'FloatingOutput' component from the ontrolDeviceBus kit to the device.

(this is valid only for pairs of triacs)

Set the channel address (see page 3).

Set the travel time of the actuator in seconds (running time from fully-open to fully-closed position).

Link the controlling logic, in the numeric range 0...100%, to the 'in' slot.

The screenshot shows the 'ontrolDeviceBus' kit in the 'Sedona Palette'. The 'Floatin' component is selected, and its configuration is shown in the right-hand pane. The configuration includes:

Property	Value	Range
Channel No	9	[0 - 255]
Status	0	
In	50	[0 - 100]
Overrider	nan	
Travel Time	90	s [0 - 255]

The 'ontrolDeviceBus' kit is also visible in the 'Sedona Palette' at the bottom of the interface, with 'FloatingOutput [88 B]' highlighted.

<sup>1</sup> also known as “raise/lower” or “3-position” control