

Using modbus drivers on Ontrol Sedona products



Various Modbus Drivers



Serial (RTU) Drivers

- `ontrolModbusMaster`
- `ontrolModbusSlave`
- `ontrolModbusSlaveSmart`

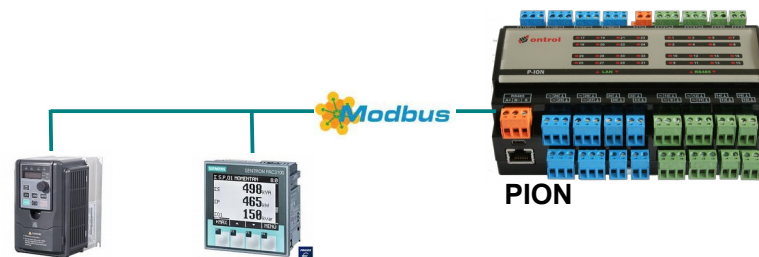
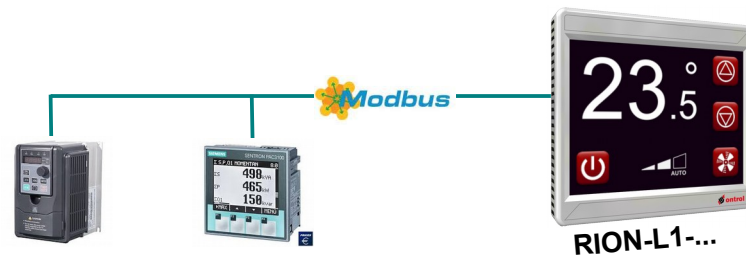
IP Drivers

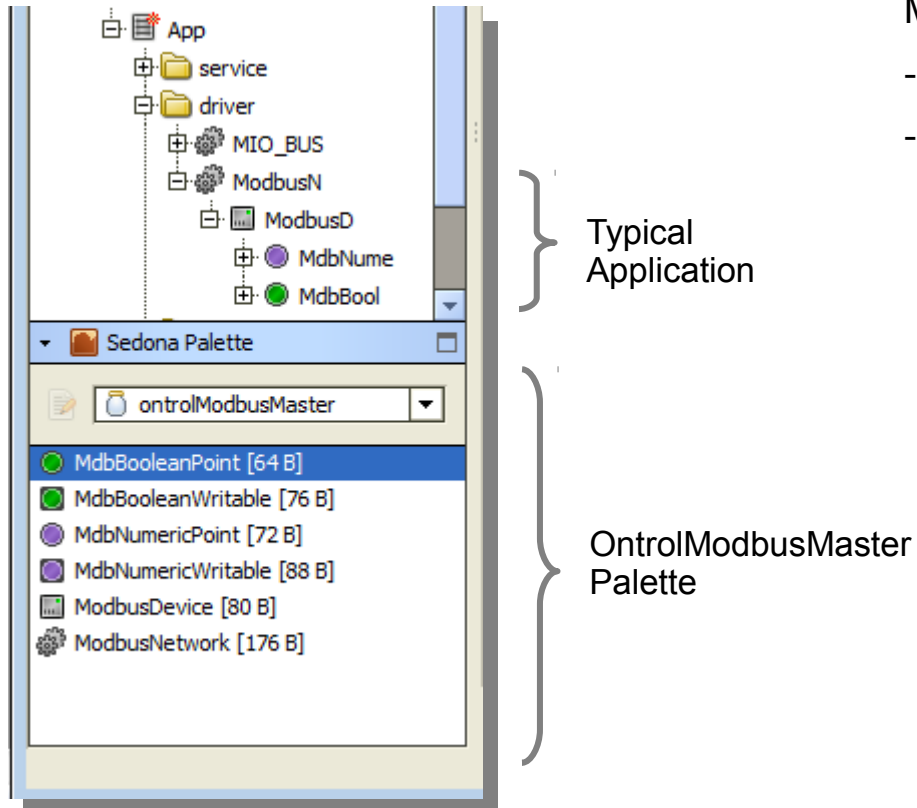
- `ontrolModbusTcpSlave`



OntronModbusMaster

Use when Ontron device is master on modbus





OBSERVE HIERARCHY

ModbusNetwork can reside anywhere but,

- ModbusDevices must go under ModbusNetwork
- Client points must go under a ModbusDevice



OntrolModbusMaster Network Properties



Serial Port number

P-ION any value

R-ION main bus : 1
R-ION aux. bus : 2
R-ION R/MIO : 250

RS485 port settings

baudrate, databits, stopbit, parity

Poll rates

(each client point has slow/normal/fast selection)

No response handling

Duration to wait for a response from slave device. Then, if no response received, number of times to retry before marking point status as fault.

ModbusN (ontrolModbusMaster::ModbusNetwork)	
<input type="radio"/> Meta	Group [1] >>
<input type="radio"/> Ping Time Sec	<input type="text" value="30"/> s
<input type="radio"/> Serial Port	<input type="text" value="0"/> [0 - 255]
<input type="radio"/> Enabled	<input checked="" type="checkbox"/> true ▼
<input type="radio"/> Ping Enabled	<input checked="" type="checkbox"/> true ▼
<input type="radio"/> Baudrate	<input type="text" value="9600"/> [1200 - 19200]
<input type="radio"/> Databits	<input type="text" value="8"/> [7 - 8]
<input type="radio"/> Stopbit	<input type="text" value="1"/> [1 - 2]
<input type="radio"/> Parity	<input type="text" value="Even"/> ▼
<input type="radio"/> Slow Rate	<input type="text" value="30"/> s [30 - 250]
<input type="radio"/> Normal Rate	<input type="text" value="5"/> s [5 - 29]
<input type="radio"/> Fast Rate	<input type="text" value="1"/> s [0 - 4]
<input type="radio"/> Max Write Time	<input type="text" value="180"/> s [0 - 250]
<input type="radio"/> Min Write Time	<input type="text" value="0"/> s [0 - 250]
<input type="radio"/> Write On Start	<input type="radio"/> false ▼
<input type="radio"/> Response Timeout	<input type="text" value="100"/> ms [50 - 3000]
<input type="radio"/> Retry Count	<input type="text" value="1"/> [0 - 10]

When ping enabled, slave devices are polled at register address 0 periodically to determine device health

maxWriteTime
Driver will repeat writes to client points with this frequency even if the value has not changed

minWriteTime
Driver will not write to an individual client point more frequently than this setting

Write to all client points on app restart?



OntrolModbusMaster Client Point Types



MdbBooleanPoint

Boolean point type polled using the modbus readCoil command

MdbBooleanWritable

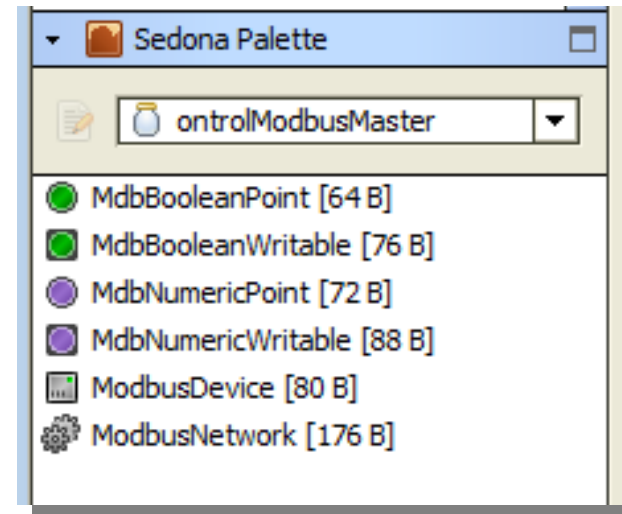
Boolean point type with an 'in' slot, allowing writes to slave device

MdbNumericPoint

Numeric point type polled using the modbus readRegister (03) command

MdbMumericWritable

Numeric point type with an 'in' slot, allowing writes to slave device





OntrolModbusMaster Client Points common properties



Properties common to all ontrolModbusMaster client points

MdbName (ontrolModbusMaster::MdbNumericPoint)	
<input type="checkbox"/> Meta	Group [1] >>
<input type="checkbox"/> Fault	<input checked="" type="radio"/> false
<input type="checkbox"/> Status	0
<input type="checkbox"/> Address	0
<input type="checkbox"/> Enabled	<input checked="" type="radio"/> true
<input type="checkbox"/> Poll Speed	Normal
<input type="checkbox"/> Out	nan

Address
Modbus register address.
Must be in decimal format.

Enabled
Register is polled only
when this property is true

Poll Speed
Fast/Medium/Slow selection.
Poll frequencies are set in
network properties

Out
Actual received value
from routine poll

Status
- Least significant bits are fault &
down as per Sedona standard
- Most significant 4 bits represent
exception message received from
slave device.



OntrolModbusMaster

Scaling for numeric client points



Scaling for numeric client points

MdbName (ontrolModbusMaster::MdbNumericPoint)

<input type="checkbox"/> <input checked="" type="radio"/> Meta	Group [1] >>
<input type="checkbox"/> <input checked="" type="radio"/> Fault	<input checked="" type="radio"/> false ▼
<input type="checkbox"/> <input checked="" type="radio"/> Status	0
<input type="checkbox"/> <input checked="" type="radio"/> Address	0
<input type="checkbox"/> <input checked="" type="radio"/> Enabled	<input checked="" type="radio"/> true ▼
<input type="checkbox"/> <input checked="" type="radio"/> Poll Speed	Normal ▼
<input type="checkbox"/> <input checked="" type="radio"/> Out	nan
<input type="checkbox"/> <input checked="" type="radio"/> Scale	1.00
<input type="checkbox"/> <input checked="" type="radio"/> Register Type	<input checked="" type="radio"/> Holding ▼

Scale

Received integer register values will be multiplied by the 'scale' property before setting the 'out' slot.

There is no 'offset' property. If you need an offset, use an additional ADD component.



OntrolModbusMaster Writable client points



Additional properties for Writable client points

MdbNum1 (ontrolModbusMaster::MdbNumericWritable)	
<input type="checkbox"/> Meta	Group [1] >>
<input type="checkbox"/> Fault	<input type="radio"/> false
<input type="checkbox"/> Status	0
<input type="checkbox"/> Address	0
<input type="checkbox"/> Enabled	<input checked="" type="radio"/> true
<input type="checkbox"/> Poll Speed	Normal
<input type="checkbox"/> Out	nan
<input type="checkbox"/> Override	nan
<input type="checkbox"/> In	0.00
<input type="checkbox"/> Scale	1.00
<input type="checkbox"/> Auto Write	<input checked="" type="radio"/> true
<input type="checkbox"/> Overriden	<input type="radio"/> false

Override
Custom components in ontrolSedonaKit module allow overriding the 'in' slot from a supervisory station

'in' slot
This is the value transmitted to the modbus register in the slave device

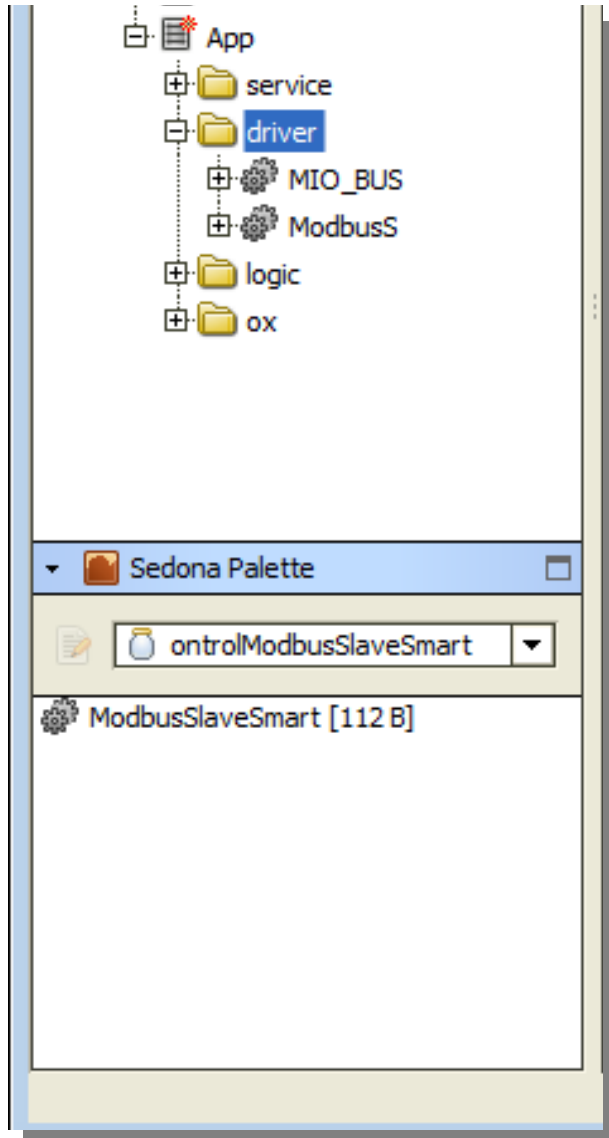


OntrolModbusSlaveSmart

Use when Ontrol device is slave on modbus



RS485 Modbus
to supervisory system



OntrolModbusSlaveSmart

A DRIVER WITH MINIMAL CONFIGURATION

- The ModbusSlaveSmart palette has only a single component.
- You only need to add the ModbusSlaveSmart network to your app.
- Each and every slot in your app becomes magically available over modbus.



OntronModbusSlaveSmart Network Properties



Serial Port number

P-ION any value

R-ION main bus : 1

R-ION aux. bus : 2

RS485 port settings

baudrate, databits, stopbit, parity

ModbusS (ontronModbusSlaveSmart::ModbusSlaveSmart)	
<input type="radio"/> Meta	Group [1] >>
<input type="radio"/> Slave Address	<input type="text" value="0"/> [1 - 247]
<input type="radio"/> Serial Port	<input type="text" value="0"/> [0 - 255]
<input type="radio"/> Enabled	<input checked="" type="radio"/> true ▼
<input type="radio"/> Baudrate	<input type="text" value="9600"/> [1200 - 19200]
<input type="radio"/> Databits	<input type="text" value="8"/> [7 - 8]
<input type="radio"/> Stopbit	<input type="text" value="1"/> [1 - 2]
<input type="radio"/> Parity	<input type="text" value="Even"/> ▼
<input type="radio"/> Scale	<input type="text" value="1.00"/>

Device Address

Scale

A single 'scale' setting applies to all numeric values transmitted on modbus.



OntrolModbusSlaveSmart Register Addresses View



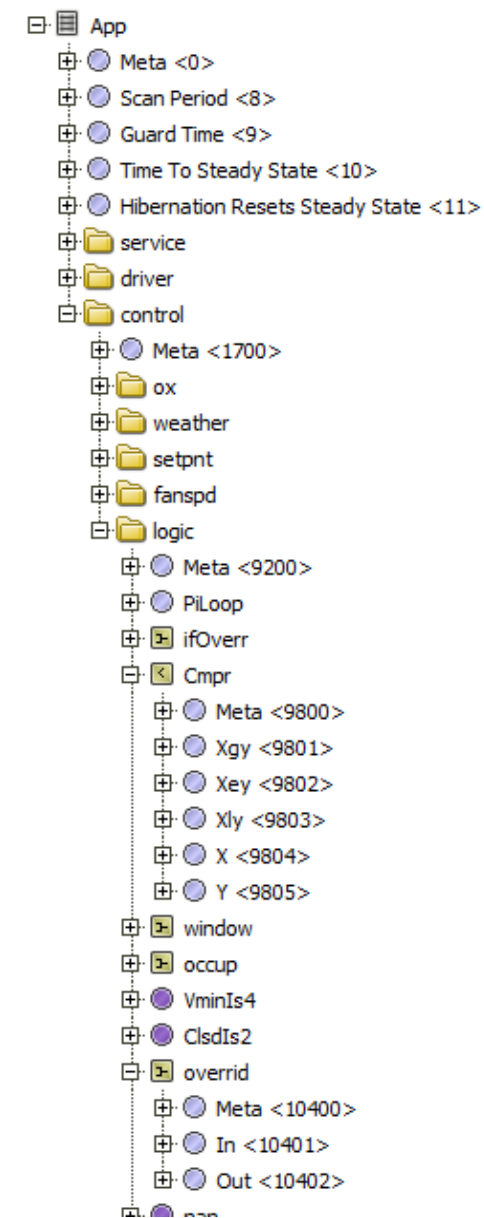
Modbus Register Addresses View

Double-clicking on the ModbusSlaveSmartNetwork shows a custom tree view of the App

Here, you can see the automatically assigned modbus register address of every slot in the app.

Requires ontrolSedonaUtil module, ver 1.0.1 or higher

(See next page for how these addresses are determined)





Addressing scheme on the ModbusSlaveSmart driver



Modbus register address = 100 x sedonaComponentId + sedonaSlotId

Determining ComponentId

Open the propertySheet view for the parent of the component.

Read componentId in the appropriate line

Property Sheet of parent

Meta	Group [1]	""
forOx	control::F2I	[forOx:32]
fanUpDn	ontrolTrigger::TrigNumericUpDown	[fanUpDn:33]
isAuto	control::Cmpr	[isAuto:34]
Const4	control::ConstFloat	[Const4:36]
fanSpd	control::ASW	[fanSpd:37]
FanSpdR	ontrolControl::RegFloat	[FanSpdR:45]
FanCmd	control::Mul3	[FanCmd:52]
Const33	control::ConstFloat	[Const33:65]

Determining SlotId

Open the slotSheet view for the component.

Count down from the top, starting at zero.

	Name	Type	Facets
0	meta	sys::int	[config]
1	out	sys::float	[readonly]
2	in1	sys::float	[]
3	in2	sys::float	[]

Slot Sheet of component

In this example,
The 'out' slot of the FanCmd component can be read via modbus register 5201.
The 'in' slot values can be read and written via modbus registers 5202 & 5203.