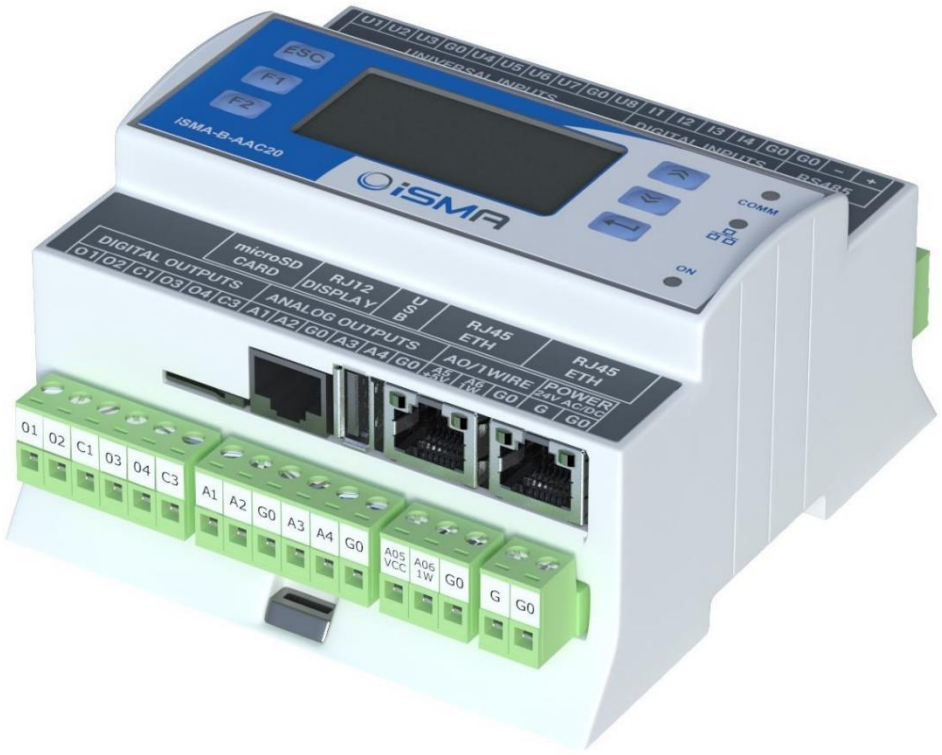
iSMA-B-AAC20

User Manual

Sedona Modbus



Global Control 5 Sp. z o.o.
Warsaw, Poland
www.gc5.pl

Table of contents

1 Introduction

This manual contains information about Modbus protocols in iSMA-B-AAC20 controller.

The iSMA-B-AAC20 controller support following types of Modbus:

- Modbus ASCII/RTU
- Modbud TCP
- Modbus TCP Slave

1.1 Revision history

Rev	Date	Description
1.0	28.08.2015	• First edition
1.3	20.04.2017	• Added Modbus type of registers addressing in Modbus TCP Slave • Added information about accuracy increasing and Resistance Register multiply in PT1000 or NI1000 input working type

2 Sedona Modbus

The Modbus protocol defines a message structure and format used in communication transactions. Modbus devices communicate using a master-slave method, in which only the master device can initiate a communications transaction. There can be only one master device on a Modbus network. All other devices must be Modbus Slaves.

WARNING! Before programming Modbus kits, please check if the latest kit version is used. The latest kits are available on GC5 support website: www.support.gc5.pl

Modbus registers

A Modbus device holds transient (real-time) data and often persistent (configuration) data in addressable ***registers***. Here, the term “registers” implies all addressable data, but this is a loose interpretation. Using Modbus nomenclature, all accessible data in a Modbus Slave is contained in the following four available groups of data flags and registers (including the Modbus Master access that is possible):

Coil status - Or simply “coils”: single-bit flags that represent the status of Digital (Boolean) Outputs of the slave, that is, On/Off output status. A Modbus Master can both read from and write to coils.

Input status - Or simply “inputs”: single-bit flags that represent the status of Digital (Boolean) Inputs of the slave, that is, On/Off output status. A Modbus Master can read (only) inputs.

Input register - Are 16-bit registers that store data collected from the field by the Modbus Slave. The Modbus Master can read (only) input registers.

Holding register - Are 16-bit registers that store general-purpose data in the Modbus Slave. The Modbus Master can both read from and write to input registers.

Modbus data addresses

A Modbus device is not required to contain all four groups of data. For example, a metering device may contain only holding registers. However, for each data group implemented, an “address convention” is used. The requests for data (made to a device) must specify a data address (and range) of interest.

Modbus data in a device is addressed as follows:

Coils — Addressed at 00000 — 0nnnn decimal, or “0x” addresses,
Inputs — Addressed at 10000 — 1nnnn decimal, or “1x” addresses,
Input Registers— Addressed at 30000 — 3nnnn decimal, or “3x” addresses,
Holding Registers — Addressed at 40000 — 4nnnn decimal, or “4x” addresses.

Note: the data addressing (at least in decimal and hex formats) is ***zero-based***, where the first instance of a data item, for example coil 1, is addressed as item number 0. As another example, holding register 108 is addressed as 107 decimal or 006B hex.

3 Installing iSMA Modbus kits

There are 4 Modbus kits in iSMA-B-AAC20 controller:

iSMA Modbus Async Network - to serve Modbus RS485 Master port of the iSMA-B-AAC20 controller,
iSMA Modules - extension of Modbus Async Netwok to serve iSMA MIX, Wireless and MINI devices using Modbus Async protocol,
iSMA Modbus TCP Network - to serve Modbus TCP Master of the iSMA-B-AAC20 device using
IP connection,
iSMA Modbus TCP Slave Network - to serve Modbus TCP Slave of the iSMA-B-AAC20 controller using IP connection.

To install Modbus kits, import the kits to the WorkPlace software (possibly as part of the package of various kits in a zip file). To do this, use an application from the Tools menu -> Sedona Installer.

After successful import of the files, upload the files to your device using the Kit Manager Application from the Sedona Tools package.

WARNING! Before programming Modbus Network, please check if the latest kit version is used. The latest kits are available on GC5 support website: www.support.gc5.pl

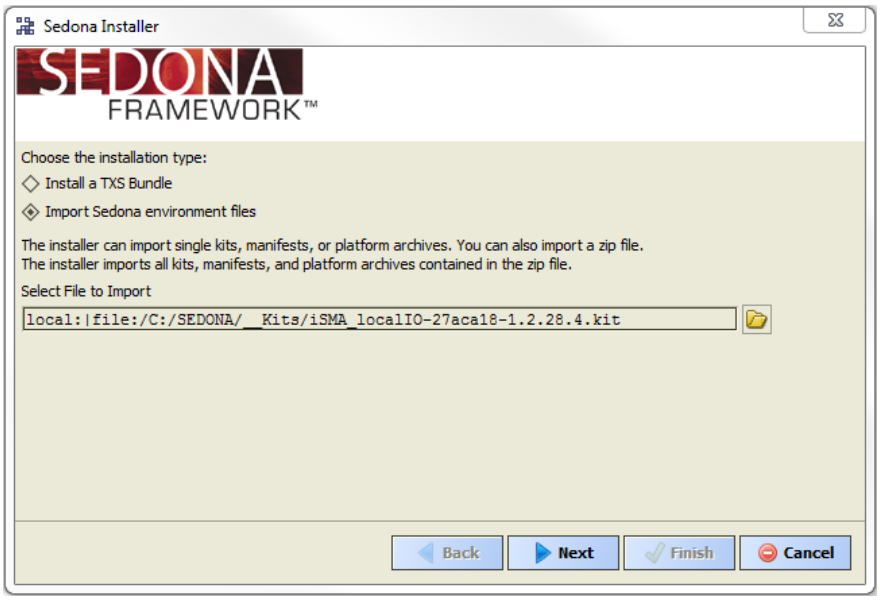


Figure 1 – Sedona Installer

4 Modbus Async Network kit

This section provides a collection of procedures to use the iSMA-B-AAC20 Modbus drivers to build networks of devices with Modbus points. The iSMA-B-AAC20 controller has one RS485 port which can be used as a Modbus RTU / ASCII Master.

Modbus Async Network kit consists of 4 types of components:

Modbus Network,
Modbus Device,
Modbus Data Points,
Modbus Points folder.

4.1 Modbus Async license and limitation

In standard license, there are available 500 data points and this number cannot be expanded. Number of available points is shown in ModbusAsyncNetwork component in slot Free Points.

WARNING! Each device and data point is counted as one point. For example to read 7 data points from 15 devices: Points number = 15 *(1 + 7) = 105

4.2 Modbus Async Network

Modbus Network is a main component which is responsible for servicing RS485 COM2 physical port. The component must be placed under Drivers folder. Modbus Network sets parameters such as communication baud rate and data format, testing etc. and keeps statistic.

The component has the following actions available under the right mouse button:

Reset Statistic – Reset network statistic value and stat counting from the beginning,
Enable/Disable – switch on/off Modbus Network.

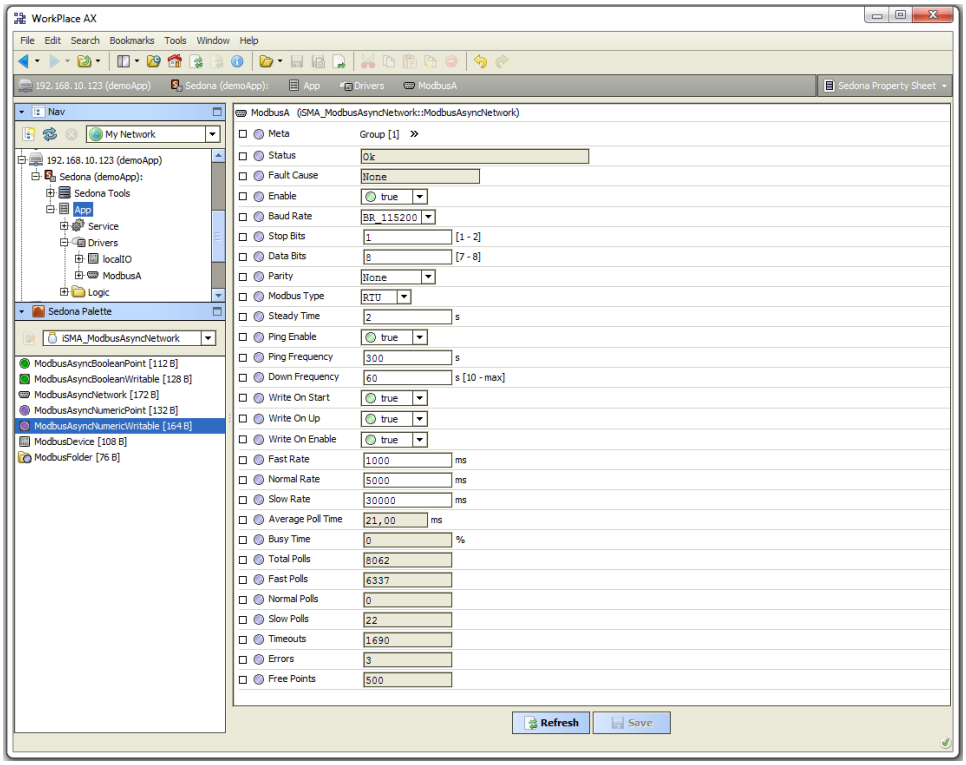


Figure 2 – Modbus Async Network component

Modbus Async Network component has the following slots:

Status – Network status, available states:
OK – Network is working properly,

Disabled – Network is disable (Slot “Enable” is in false),

OK some device/point down – error in device or points.

Fault Cause – Fault cause description,

Enable – this option switch on or switch off Modbus Network
(„true”- Network enable, „false”- Network disable),

Steady Time – Network start up delay time after power up or reset,

Baud Rate – Modbus RS485 port baud rate, available options:
2400, 4800, 9600, 19200, 38400, 57600, 115200 bps,

Stop Bits – Stop bit definition, available options: 1-bit, 2-bits,

Data Bits – Data bits definition, available options: 7-bits or 8-bits,

Parity – Parity bit definition, available options: None, Odd, Even, Always1, Always0,

Modbus Type – Modbus type definition , available options: RTU or ASCII,

Ping Enable – Enable devices connection testing function,

Ping Frequency – Time between testing message to check device connection,

Down Frequency – Time between testing message for devices or points which have got status down,

Write On Start – Do write action in device „Writable” components in Modbus Network after reset or power up,

Write On Up - Do write action in device „Writable” components in Modbus Network after restore the connection with Modbus device,

Write On Enable - Do write action in device „Writable” components in Modbus Network after enable the device,

Fast Rate – Time between messages in „Fast” mode poll frequency,

Normal Rate – Time between messages in „Normal” mode poll frequency,

Slow Rate – Time between messages in „Slow” mode poll frequency,

Average Poll Time – Average time for sending/receive one message,

Busy Time – Percentage of Modbus Network usage,

Total Polls – Total number of messages,

Fast Polls – Number of messages in sending „Fast” mode,

Normal Polls – Number of messages in sending „Normal” mode,

Slow Polls – Number of messages in sending „Slow” mode,

Timeouts – Number of lost messages, difference between sending and receive messages,

Errors – Number of error messages (for example with wrong CRC)

Free points – Number of available physical points in Modbus Network.

4.3 Modbus Async Device

ModbusAsyncDevice is a component which is responsible for servicing physical device connected to Modbus Network. The device as Modbus Master to all other Modbus devices on the attached RS485 port. Each device is represented by a Modbus Device, and has a unique Modbus address (1 to 247), as well as other Modbus config data and starting addresses for Modbus data items (coils, inputs, input registers, holding registers). The component has Ping action available under the right mouse button, which sends test message to device to check device status. Each ModbusAsyncDevice has a “Ping Address” container slot with 3 properties slots (Address Format, Ping Address Reg, Ping Type). These properties specify a particular data address (either input register or holding register) to use as the device status test (meaning “Monitor” ping requests). Ping requests are generated at the network-level by the configurable network monitor (ModbusNetwork -> Ping Enabled). When enabled, a network’s monitor periodically pings (queries) this address. While receiving Any response from the device, including an exception response, this is considered proof of communication, and the Modbus client device is no longer considered “down” if it had been previously marked “down”. The component has Ping action available under the right mouse button, which sends test message to device to check device status.

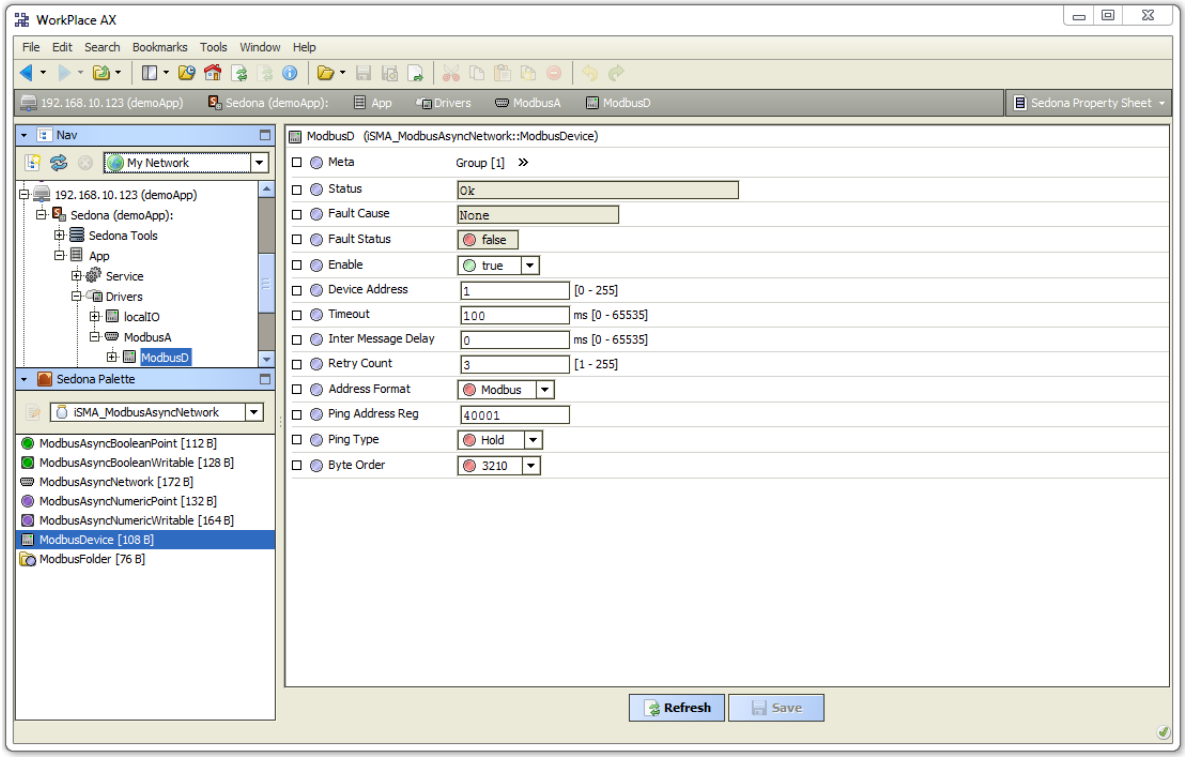


Figure 3 - Modbus Async Device component

Modbus Async Device has the following slots:

Status – Device actual status (read Only), can take the following states:
OK – Device is working properly,

Disable – Device is disable (Slot “Enable” is in false),

Down – Device is not available,

Ok, some points down/error – error in points reading,

Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,

Fault Status – device error status (true – device communication error),

Enable – Enable/Disable device,

Device Address – Modbus device physical address (0 – network broadcast address, 1-248 addressing range),

Timeout – Max. device response time from device request,

Inter Message Delay – Time between messages sending to device,

Retry Count – Max. error messages (CRC error, lost messages),

Address Format – Modbus address format (Modbus, Decimal),

Ping Address Reg – Any register number Input or Holding type, which will be reading for device connection test,

Ping Type – Tested register type: Input/Holding,

Byte Order – Byte order reading 32-bit: 3210 (Big endian), 1032 (Little endian).

4.4 Modbus data points

In Modbus protocol each device has implemented Modbus table. Sedona has 5 components to read/write data from this table:

Boolean Point – Read Boolean value (Modbus command 0x02),
Boolean Writable – Read/write Boolean value (Modbus command 0x05),
Numeric Point – Read numeric value (Modbus commands: Input - 0x04, Holding – 0x03),
Numeric Writable – Read/write numeric value (Modbus commands: 16-bits Int, SInt - 0x06, 32-bits Long, SLong, Float – 0x16),
Numeric Multi Point – Read up to 8 16-bits registers (Modbus command 0x16).

4.4.1 Modbus Async Boolean Point

Modbus Async Boolean Point is a component which is responsible for reading Boolean values from the device. The component has Read action available under the right mouse button, which forces the reading of the point.

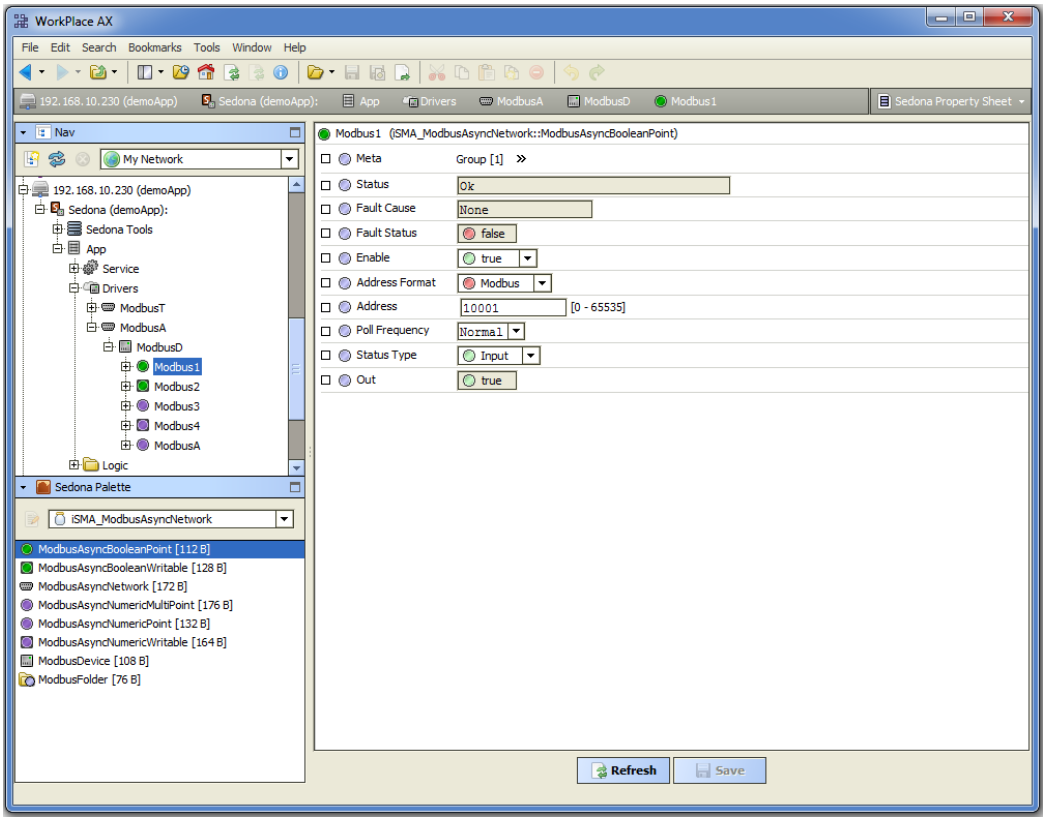


*Figure 4 - Modbus Async Boolean Point component*Modbus Async

Boolean Point component has the following slots:

Status – Point status, available states:
OK – Point is working properly,

Disabled – Point is disable (Slot “Enable” is in false),
Down/Timeout – Point is not available,
Device Down – Device is not available,
Wrong address format – incorrect address format according to address format setting slot,
Device disabled - Device is disabled,
Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,
Fault Status – point error status (true – point read error),
Enable – Point enable/disable („true”- Point enable, „false”- Point disable),
Address Format – Register address format, available options: Modbus, Decimal,
Address – Register address,
Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,
Status Type – Type of reading register, available options: Input – 0x02, Coil – 0x01,
Out – Current value of read registry.

4.4.2 Modbus Async Boolean Writable

Modbus Async Boolean Writable is a component which is responsible for sending and reading Boolean values from device.

The component has the following actions available under the right mouse button:

Set True/Set False – Write value to slot In and sends it to the device (not active when slot In has connected link),
Write – send value from slot In to the device,
Read – read value from device and send to slot Out.

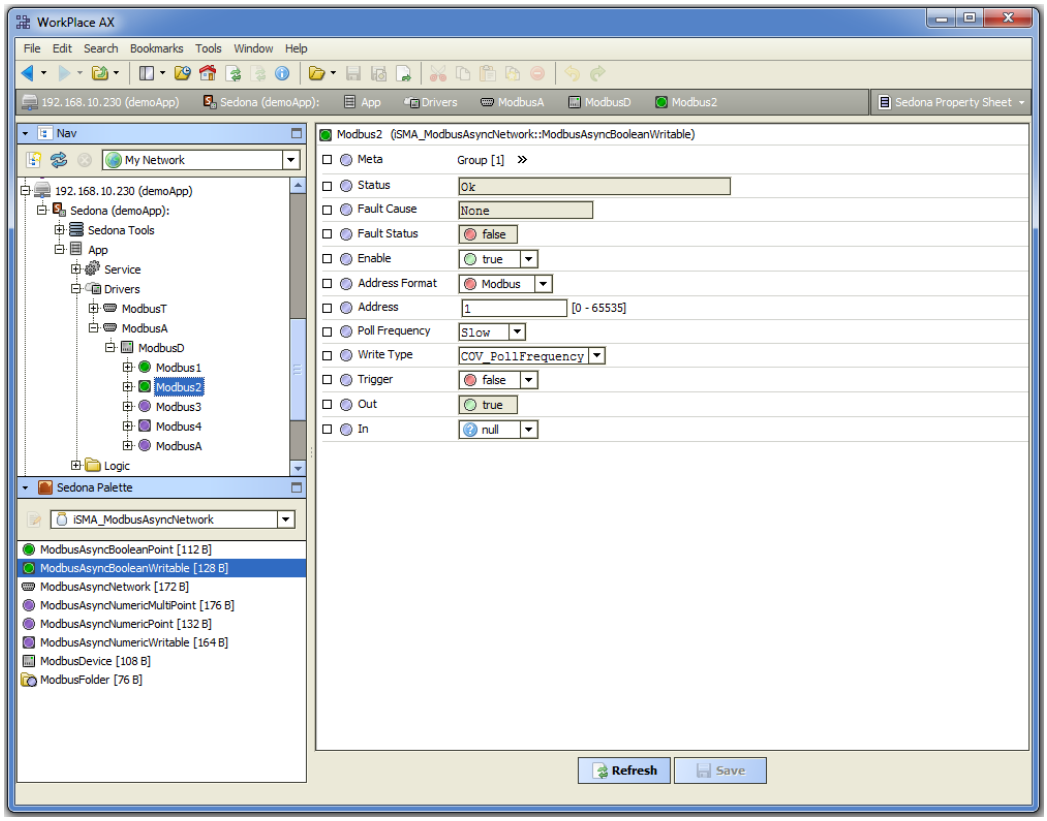


Figure 5 - Modbus Async Boolean Writable component

Modbus Async Boolean Writable component has the following slots:

Status – Point status, available states:
OK – Point is working properly,

Disabled – Point is disable (Slot “Enable” is in false),

Down/Timeout – Point is not available,

Device Down – Device is not available,

Wrong address format – Incorrect address format according to address format setting slot,

Device disabled - Device is disabled,

Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,

Fault Status – Point error status (true – point read/write error),

Enable – Point enable/disable („true”- Point enable, „false”- Point disable),

Address Format – Register address format, available options: Modbus, Decimal,

Address – Register address,

Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,

Write Type – Writing mode, available options: COV – only on input change, COV_PollFrequency – on input change and periodically, PollFrequency - only periodically, COV_LinkSet (Link-back forward triggers by COV)

Trigger - Remote force sending trigger (on rising edge),

Out – Output slot, current value of read/write registry,

In – Input slot.

4.4.3 Modbus Async Numeric Point

Modbus Async Numeric Point is a component which is responsible for reading numeric values from the device. The component has Read action available under the right mouse button, which forces the reading of the point.

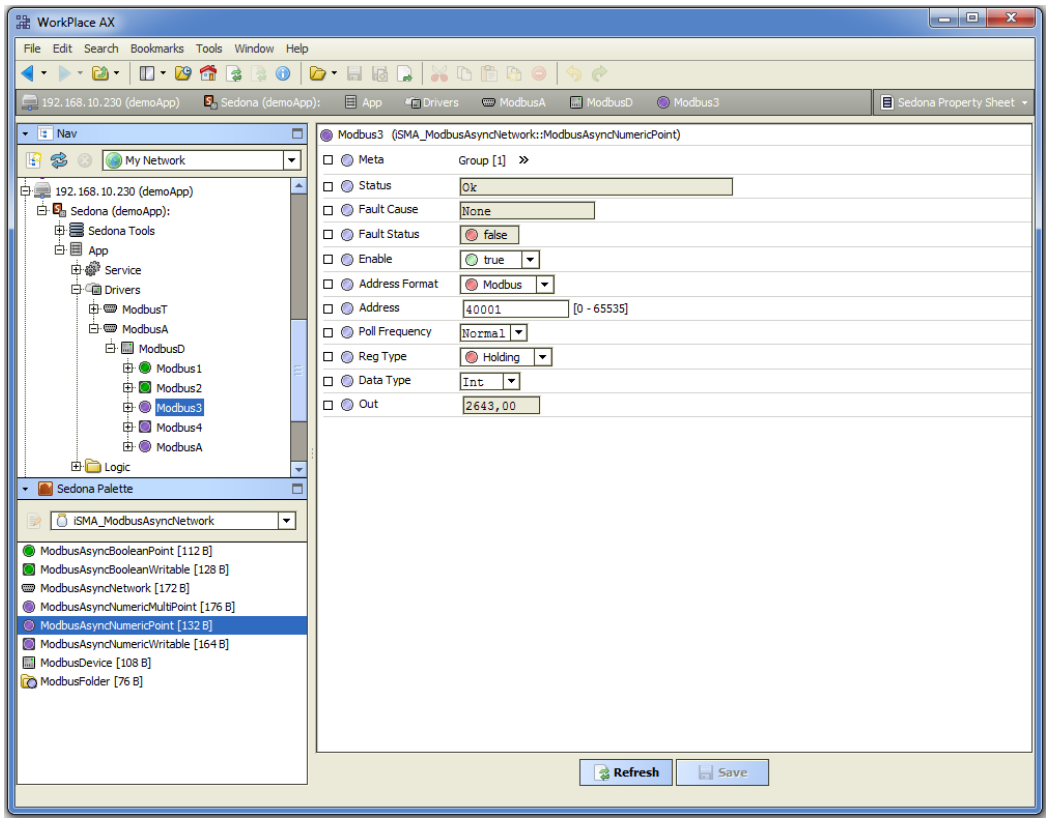


Figure 6 – Modbus Async Numeric Point component

Modbus Async Numeric Point component has the following slots:

Status – Point status, available states:
OK – Point is working properly,
Disabled – Point is disable (Slot “Enable” is in false),
Down/Timeout – Point is not available,
Device Down – Device is not available,
Wrong address format – Incorrect address format according to address format setting slot,
Device disabled - Device is disabled,
Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,
Fault Status – Point error status (true – point read error),
Enable – Point enable/disable („true”- Point enable, „false”- Point disable),

Address Format – Register address format, available options: Modbus, Decimal,
Address – Register address,
Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,
Reg Type – Type of reading register, available options: Input – 0x04, Holding – 0x03,
Data Type – Reading registry data type, available options: Int – 16-bits, Long – 32-bits, Float – 32-bits float-point, SInt – 16-bits with sign, Slong – 32-bits with sign,
Out – Current value of read registry.

4.4.4 Modbus Async Numeric Writable

Modbus Async Numeric Writable is a component which is responsible for sending and reading numeric values from the device.

The component has the following actions available under the right mouse button:

Set – write value to slot In and sends it to the device,
Write – send value from slot In to device,
Read – read value from device and sends it to slot Out.

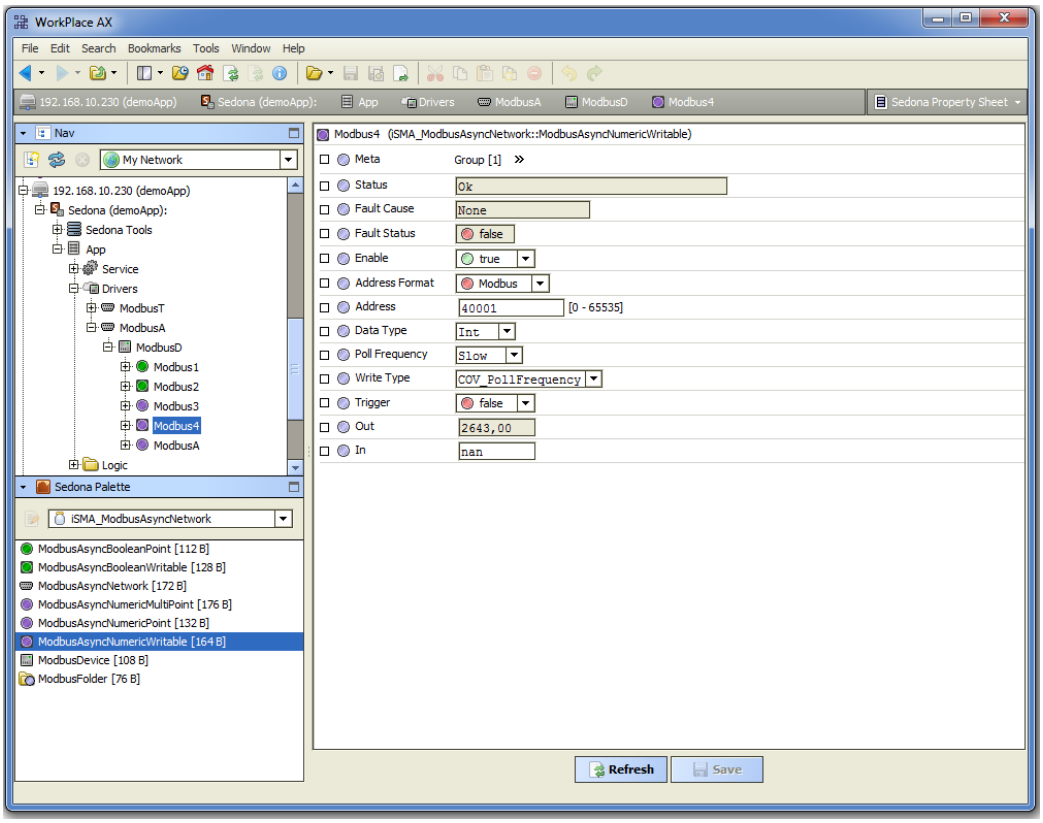


Figure 7 - Modbus Async Numeric Writable component

Modbus Async Numeric Writable component has the following slots:

Status – Network status, available states:
OK – Point is working properly,

Disabled – Point is disable (Slot “Enable” is in false),

Down/Timeout – Point is not available,

Device Down – Device is not available,

Wrong address format – Incorrect address format according to address format setting slot,

Device disabled - Device is disabled,

Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,

Fault Status – Point error status (true – point read/write error),

Address Format – Register address format, available options: Modbus, Decimal,

Address – Register address,

Data Type – Read/write registry data type, available options: Int – 16-bits, Long – 32-bits, Float – 32-bits float-point, SInt – 16-bits with sign, Slong – 32-bits with sign, IntF16- use Function 16, SIntF16 – use Function 16 (Function 16 – modbus function for sending one register)

Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,

Write Type – Writing mode, available options: COV – only on input change, COV_PollFrequency – on input change and periodically, PollFrequency - only periodically, COV_LinkSet (Link-back forward triggers by COV)

Trigger – Remote force sending trigger (on rising edge),

Out – Output slot, current value of device registry,

In – Input slot.

4.4.5 Modbus Async Numeric Multi Point

Modbus Async Numeric Multi Point is a component which is responsible for reading up to 8 16-bits registers from the device in one message. The component use 0x16 Modbus command. The component has Read action available under the right mouse button, which forces the reading of the point.

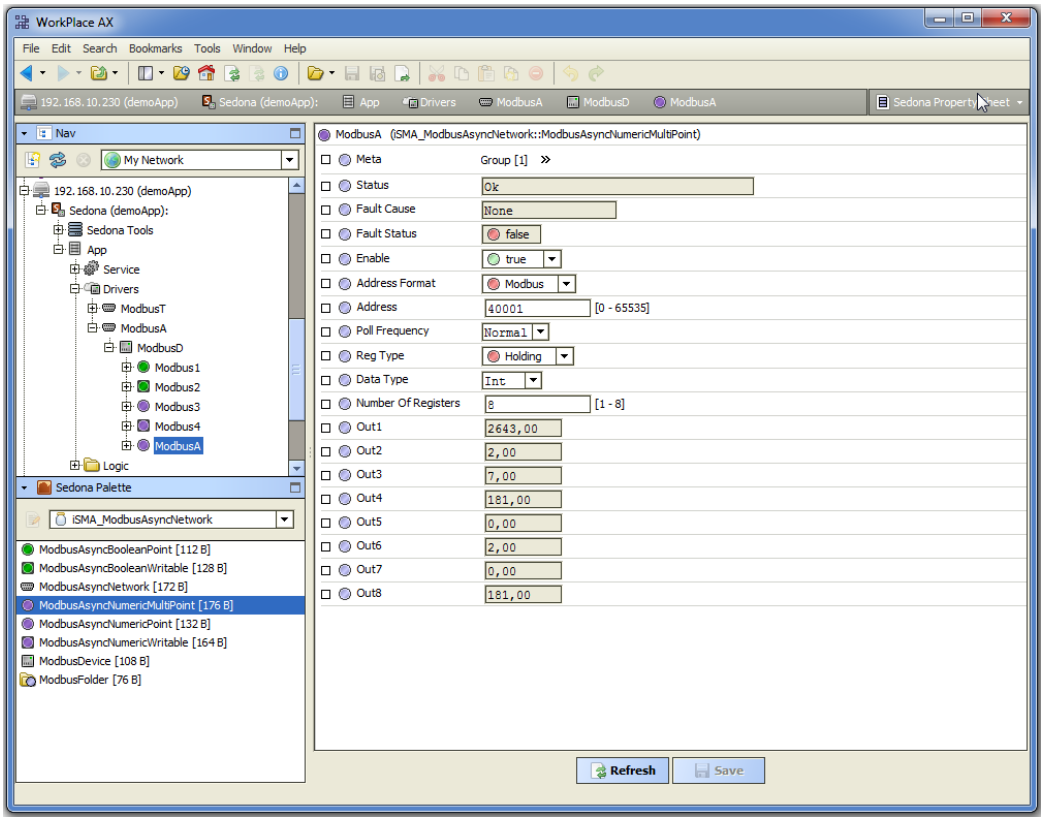


Figure 8 - Modbus Async Numeric Multi Point component

Modbus Async Numeric Multipoint component has the following slots:

Status – Point status, available states:
OK – Point is working properly,
Disabled – Point is disable (Slot “Enable” is in false),
Down/Timeout – Point is not available,
Device Down – Device is not available,
Wrong address format – Incorrect address format according to address format setting slot,
Device disabled - Device is disabled,
Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,
Fault Status – Point error status (true – point read error),
Enable – Point enable/disable („true”- Point enable, „false”- Point disable),
Address Format – Register address format, available options: Modbus, Decimal,
Address – Register address,
Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,
Reg Type – Type of reading register, available options: Input – 0x04, Holding – 0x03,
Data Type – Read data type: Int (unsigned values) Sint (signed values),
Number Of Registers – Number of reading register in one messages,
Out – Current value of read registry.

4.5 Modbus folder

ModbusFolder is a component which groups and organizes Modbus points components. Because of the Sedona components name are limited to 7 characters, ModbusFolder has Description Slot where we can us up to 32 characters.

5 Modbus TCP Network kit

The iSMA-B-AAC20 controller has implemented Modbus TCP protocol. It means that controlled as a Master device can read/write data to Slave devices using IP connection.

5.1 Modbus TCP license and limitation

In controller without Modbus license there are available 15 data points. Licence can be expanded up to 500 points by purchasing Modbus license. Number of available points is shown in MoodbusTcpNetwork component in slot Free Points.

WARNING! Each device and data point is counted as one point. For example to read 4 data points from 2 devices: Points number = 2 *(1 + 4) = 10

5.2 Modbus TCP Network

Modbus TCP network is a main component which is responsible for servicing IP communication to slave devices. The component must be placed under Drivers folder. ModbusNetwork sets parameters such as communication baud rate and data format, testing etc. and keeps statistic.

The component has the following actions available under the right mouse button:

Reset Statistic – Reset network statistic value and stat counting from the beginning
Enable/Disable – switch on/off Modbus TCP network.

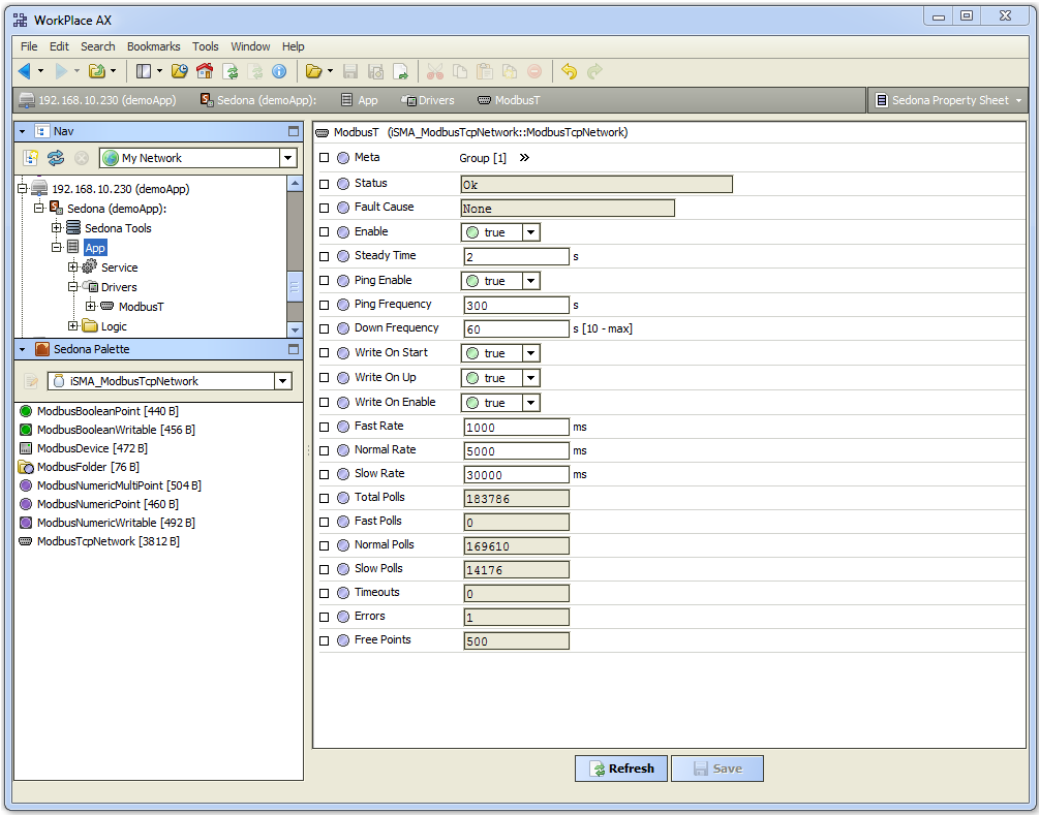


Figure 9 - Modbus TCP Network component

Modbus TCP Network component has following slots:

Status – Network status, available states:
OK – Network is working properly,

Disabled – Network is disable (Slot “Enable” is in false),

OK some device/point down – error in device or points.

Fault Cause – Fault cause description,

Enable –Option switch on or switch off Modbus network
(„true”- Network enable, „false”- Network disable),

Steady Time – Network start up delay time after power up or reset,

Ping Enable – Enable devices connection testing function,

Ping Frequency – Time between testing message to check device connection,

Down Frequency – Time between testing message for devices or points which have got status down,

Write On Start – Do write action in device „Writable” components in Modbus network after reset or power up,

Write On Up - Do write action in device „Writable” components in Modbus network after restore the connection with Modbus device,

Write On Enable - Do write action in device „Writable” components in Modbus network after enable the device,

Fast Rate – Time between messages in „Fast” mode poll frequency,

Normal Rate – Time between messages in „Normal” mode poll frequency,

Slow Rate – Time between messages in „Slow" mode poll frequency,
Average Poll Time – Average time for sending/receiving one message,
Busy Time – Percentage of Modbus network usage,
Total Polls – Total number of messages,
Fast Polls – Number of messages in sending „Fast" mode,
Normal Polls – Number of messages in sending „Normal" mode,
Slow Polls – Number of messages in sending „Slow" mode,
Timeouts – Number of lost messages, difference between sending and receiving messages,
Errors – Number of error messages (for example with wrong CRC)
Free points – Number of available physical points in Modbus Network.

5.3 Modbus TCP Device

ModbusDeviece is a component which is responsible for servicing physical Modbus TCP slave devices. Each Modbus device is represented by an IP address, port number (default for Modbus 502) and Device address (1 to 247).

The component has Ping action available under the right mouse button, which sends test message to device to check device status. Each ModbusDevice has a "Ping Address" container slot with 3 properties slots (Address Format, Ping Address Reg, Ping Type). These properties specify a particular data address (either input register or holding register) to use as the device status test (meaning "Monitor" ping requests). Ping requests are generated at the network-level by the configurable network monitor (ModbusNetwork -> Ping Enabled). When enabled, a network's monitor periodically pings (queries) this address. While receiving Any response from the device, including an exception response, this is considered proof of communication, and the Modbus client device is no longer considered "down" if it had been previously marked "down". The component has Ping action available under the right mouse button, with sends test message to device to check device status.

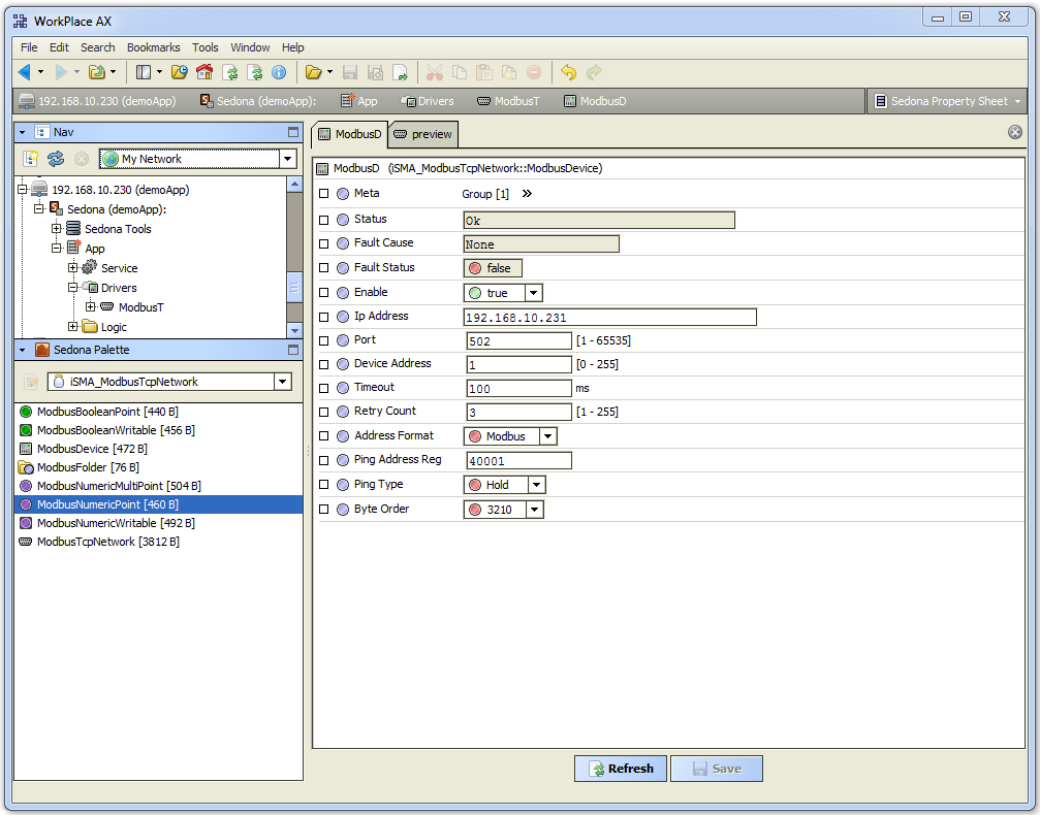


Figure 10 - Modbus TCP Device component

Modbus TCP Device has the following slots:

Status – Device actual status (read Only), can take the following states:
OK – Device is working properly,

Disable – Device is disable (Slot “Enable” is in false),

Down – Device is not available,

Ok, some points down/error – error in points reading,

Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,

Fault Status – Device error status (true – device communication error),

Enable – Enable/Disable device,

IP Address – Slave device (gateway) IP address,

Port – Slave device (gateway) Modbus TCP port number (default 502),

Device Address – Modbus device address (0 – broadcast, 1-248 addressing range),

Timeout – Max. device response time from device request,

Inter Message Delay – Time between messages sending to device,

Retry Count – Max. error messages (CRC error, lost messages),

Address Format – Modbus address format (Modbus, Decimal),

Ping Address Reg – Any register number Input or Holding, for device connection test,

Ping Type – Tested register type: Input/Holding,

Byte Order – Byte order reading 32-bit: 3210 (Big endian), 1032 (Little endian).

5.4 Modbus TCP data points

In Modbus protocol each device has implemented Modbus table. Sedona has 5 components to read/write data from this table:

Boolean Point – Read Boolean value (Modbus command 0x02),
Boolean Writable – Read/write Boolean value (Modbus command 0x05),
Numeric Point – Read numeric value (Modbus commands: Input - 0x04, Holding – 0x03),
Numeric Writable – Read/write numeric value (Modbus commands: 16-bits Int, SInt - 0x06, 32-bits Long, SLong, Float – 0x16),
Numeric Multi Point – Read up to 8 16-bits registers (Modbus command 0x16).

5.4.1 Modbus TCP Boolean Point

Modbus TCP Boolean Point is a component which is responsible for reading Boolean values from the device. The component has Read action available under the right mouse button, which forces the reading of the point.

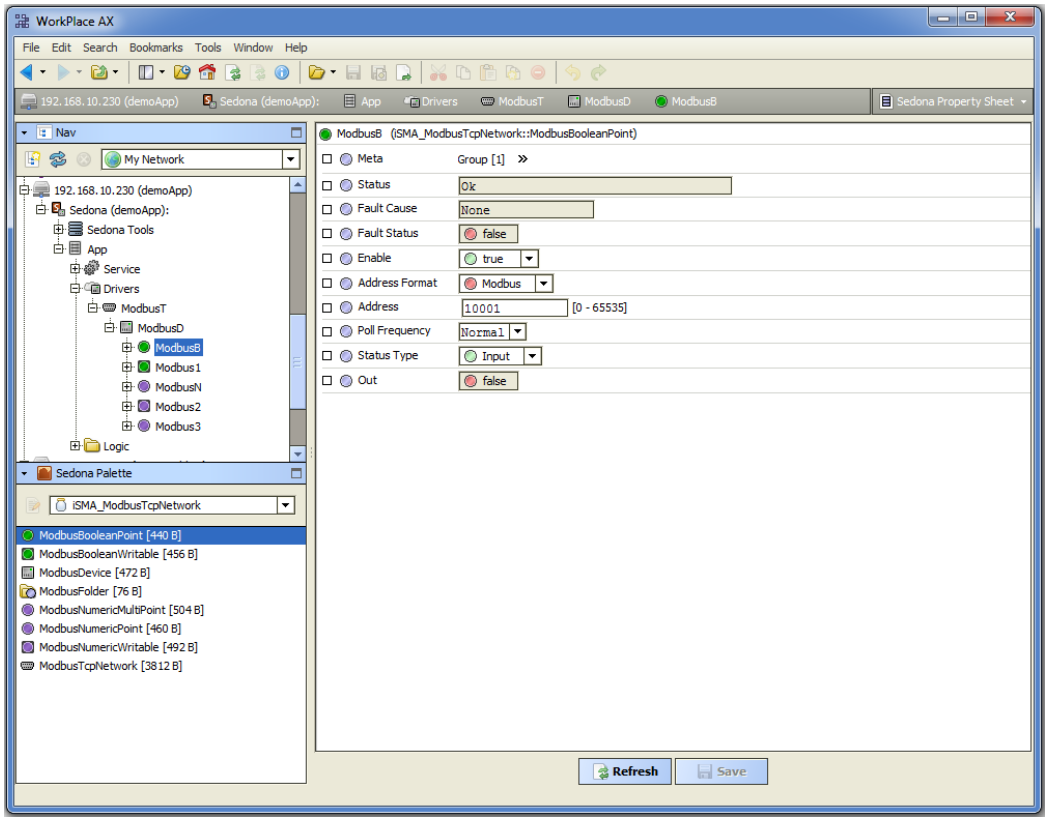


Figure 11 - Modbus TCP Boolean Point component

Modbus TCP Boolean Point component has the following slots:

Status – Network status, available states:
OK – Point is working properly,

Disabled – Point is disable (Slot “Enable” is in false),

Down/Timeout – Point is not available,
Device Down – Device is not available,
Wrong address format – Incorrect address format according to address format setting slot,
Device disabled - Device is disabled,
Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,
Fault Status – Point error status (true – point read error),
Enable – Point enable/disable („true”- Point enable, „false”- Point disable),
Address Format – Register address format, available options: Modbus, Decimal,
Address – Register address,
Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,
Status Type – Type of reading register, available options: Input – 0x02, Coil – 0x01,
Out – Current value of read registry.

5.4.2 Modbus TCP Boolean Writable

Modbus TCP Boolean Writable is a component which is responsible for sending and reading Boolean values from device.

The component has the following actions available under the right mouse button:

Set True/Set False – Write value to slot In and sends it to the device (not active when slot In have connected link),
Write – send value from slot In to the device,
Read – read value from device and send to slot Out.

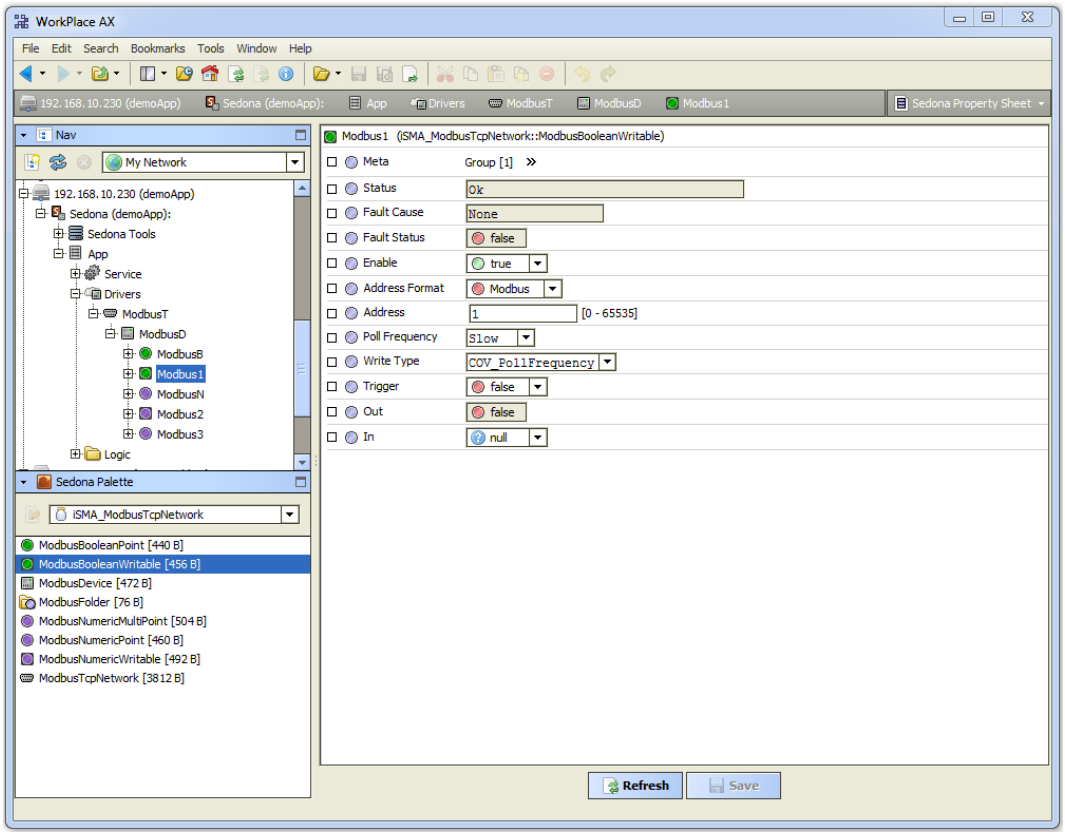


Figure 12 – Modbus TCP Boolean Writable component

Modbus Boolean Writable component has the following slots:

Status – Point status, available states:
OK – Point is working properly,

Disabled – Point is disable (Slot “Enable” is in false),

Down/Timeout – Point is not available,

Device Down – Device is not available,

Wrong address format – Incorrect address format according to address format setting slot,

Device disabled - Device is disabled,

Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,

Fault Status – Point error status (true – point read/write error),

Enable – Point enable/disable („true”- Point enable, „false”- Point disable),

Address Format – Register address format, available options: Modbus, Decimal,

Address – Register address,

Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,

Write Type – Writing mode, available options: COV – only on input change, COV_PollFrequency – on input change and periodically, PollFrequency - only periodically, COV_LinkSet (Link-back forward triggers by COV)

Trigger – Remote force sending trigger (on rising edge),

Out – Output slot, current value of read/write registry,

In – Input slot.

5.4.3 Modbus TCP Numeric Point

ModbusClientNumericPoint is a component which is responsible for reading numeric values from the device. The component has Read action available under the right mouse button, which forces the reading of the point.

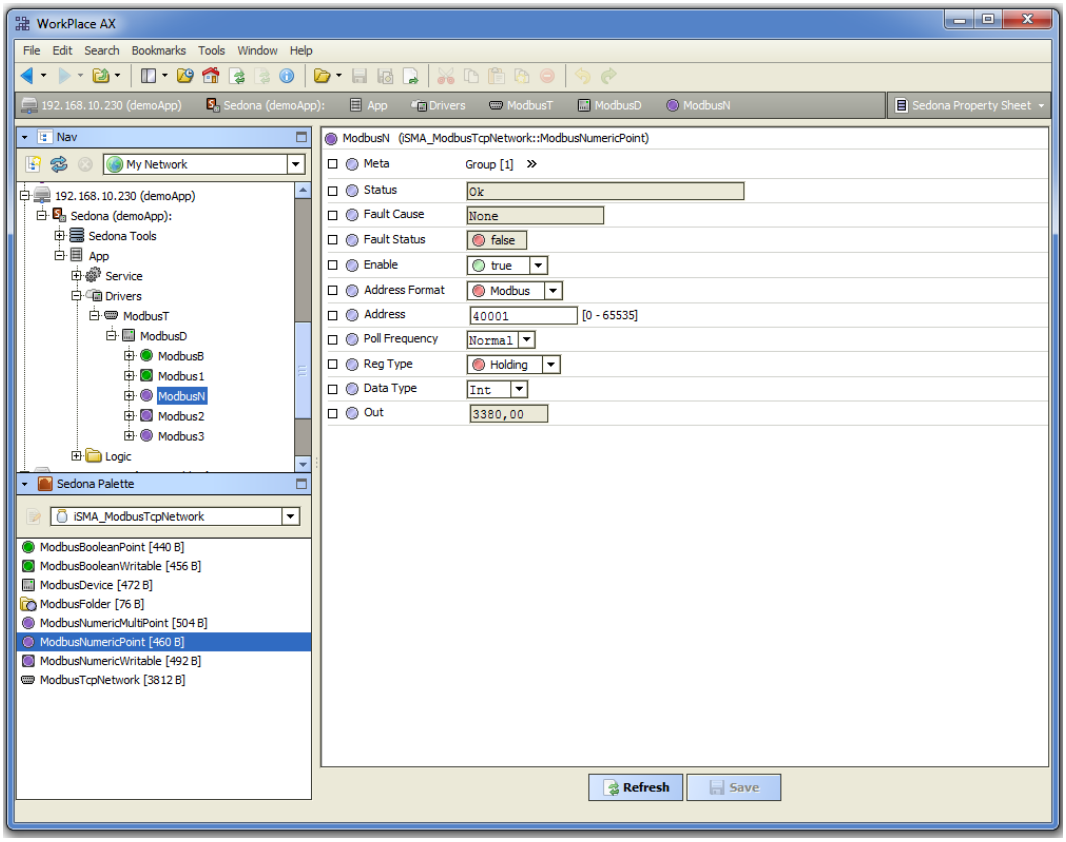


Figure 13 - Modbus TCP Numeric Point component

Modbus TCP Numeric Point component has the following slots:

Status – Network status, available states:
OK – Point is working properly,
Disabled – Point is disable (Slot “Enable” is in false),
Down/Timeout – Point is not available,
Device Down – Device is not available,
Wrong address format – Incorrect address format according to address format setting slot,
Device disabled - Device is disabled,
Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,
Fault Status – Point error status (true – point read error),
Enable – Point enable/disable („true”- Point enable, „false”- Point disable),

Address Format – Register address format, available options: Modbus, Decimal,
Address – Register address,
Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,
Reg Type – Type of reading register, available options: Input – 0x04, Holding – 0x03,

Data Type – Reading registry data type, available options: Int – 16-bits, Long – 32-bits, Float
– 32-bits folat-point, SInt – 16-bits with sign, Slong – 32-bits with sign,
Out – Current value of read registry.

5.4.4 Modbus TCP Numeric Writable

ModbusClientNumericWritable is a component which is responsible for sending and reading Numeric values from device.

The component has the following actions available under the right mouse button:

Set– Write value to slot In and sends it to the device,
Write – send value from slot In to device,
Read – read value from device and sends it to slot Out.

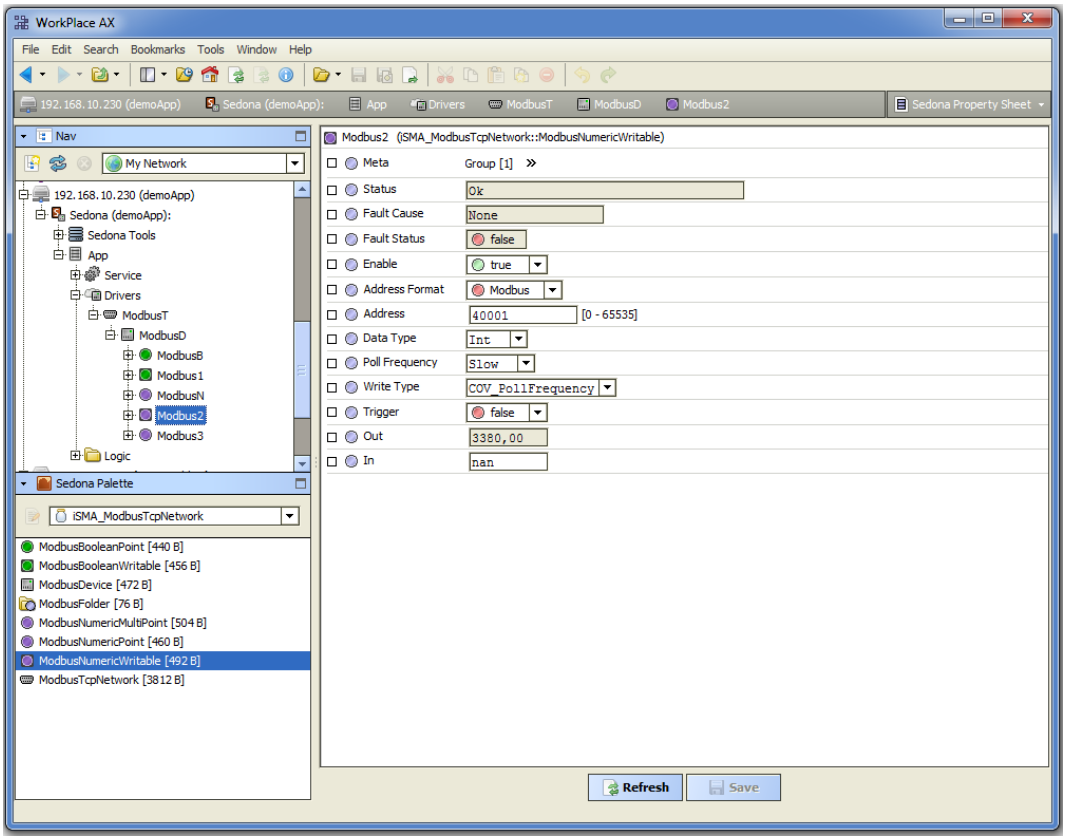


Figure 14 - Modbus TCP Numeric Writable component

Modbus Numeric Writable component has following slots:

Status – Network status, available states:
OK – Point is working properly,

Disabled – Point is disable (Slot “Enable” is in false),

Down/Timeout – Point is not available,

Device Down – Device is not available,

Wrong address format – Incorrect address format according to address format setting slot,

Device disabled - Device is disabled,

Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,

Fault Status – Point error status (true – point read/write error),

Address Format – Register address format, available options: Modbus, Decimal,

Address – Register address,

Data Type – Read/write registry data type, available options: Int – 16-bits, Long – 32-bits, Float – 32-bits float-point, SInt – 16-bits with sign, Slong – 32-bits with sign, IntF16- use Function 16, SIntF16 – use Function 16 (Function 16 – modbus function for sending one register)

Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,

Write Type – Writing mode, available options: COV – only on input change,
COV_PollFrequency – on input change and periodically, PollFrequency - only periodically, COV_LinkSet (Link-back forward triggers by COV)

Trigger – Remote force sending trigger (on rising edge),

Out – Output slot, current value of device registry,

In – Input slot.

5.4.5 Modbus TCP Numeric Multi Point

Modbus Async Numeric Milti Point is a component which is responsible for reading up to 8 registers from the device in one message. The component use 0x16 Modbus command The component has Read action available under the right mouse button, which forces the reading of the point.

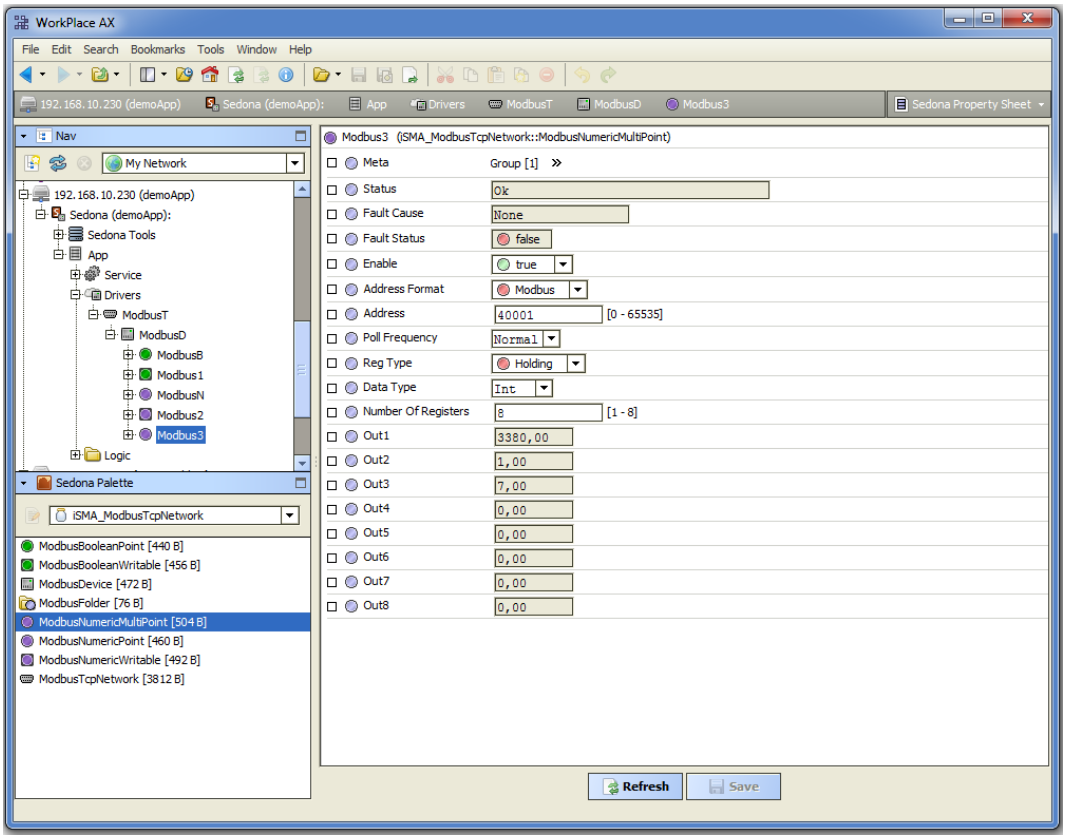


Figure 15 - Modbus TCP Numeric Multipoint property sheet view

Modbus TCP Numeric Multipoint component has following slots:

Status – Point status, available states:
OK – Point is working properly,
Disabled – Point is disable (Slot “Enable” is in false),
Down/Timeout – Point is not available,
Device Down – Device is not available,
Wrong address format – Incorrect address format according to address format setting slot,
Device disabled - Device is disabled,
Network disabled – Modbus Network is disabled.

Fault Cause – Fault cause description,
Fault Status – Point error status (true – point read error),
Enable – Point enable/disable („true”- Point enable, „false”- Point disable),
Address Format – Register address format, available options: Modbus, Decimal,
Address – Register address,
Poll Frequency – Reading poll frequency, available options: Fast, Normal, Slow,
Reg Type – Type of reading register, available options: Input – 0x04, Holding – 0x03,
Data Type – Read data type: Int (unsigned values) Sint (signed values),
Number Of Registers – Number of reading register in one messages,
Out – Current value of read registry.

5.5 Modbus folder

ModbusFolder is a component which groups and organizes Modbus points components. Because of Sedona components name are limited to 7 characters, ModbusFolder has Description Slot where we can us up to 32 characters.

6 Modbus TCP Slave Network kit (IP Module)

The controller has built-in Modbus TCP/IP Server (Modbus TCP Slave Device) application which is always active. The controller has built-in register table to read/write data. Addresses from 0 to 999 (decimal numeration) are reserved for controller registers. To see list of registers and registers parameters go to iSMA-B-AAC20 Modbus Table chapter. Addresses from 1000 – 2999 are free to use and can be adopted in user application.

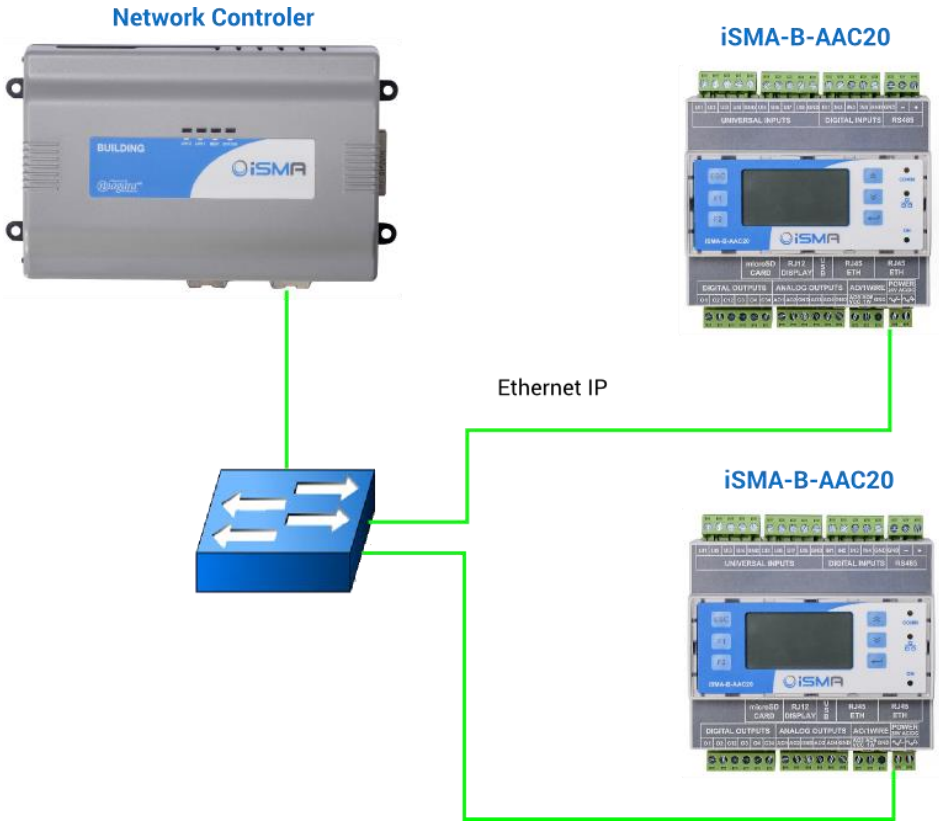


Figure 16 - Modbus TCP topology

6.1 Modbus TCP Slave Network

Modbus TCP Slave is always enable and to read controller registers it doesn't have to be configured. Modbus TCP Slave network is use only for changing parameters (parameters can by changed also from controller configuration web page) and to set up user registers.

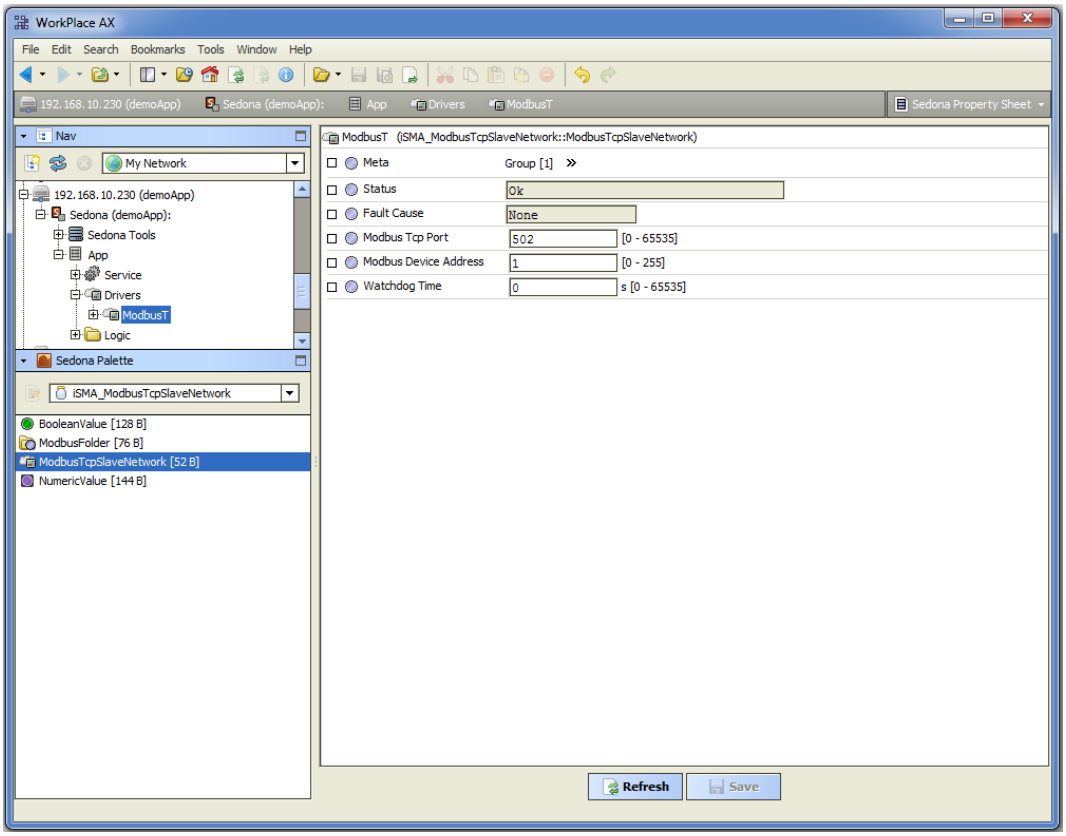


Figure 17 - Modbus TCP Slave network component

Modbus TCP Slave Network has the following slots:

Status – Network status,
Fault Cause – Fault cause description,
Modbus TCP Port – Modbus TCP Port number (default 502),
Modbus Device Address – Controller Modbus address,
Watchdog Time – Time between receive valid messages after which controller will set default values on outputs, value 0 disable this function.

6.2 Modbus TCP Slave data points

Modbus TCP Slave data points are served by two components placed under ModbusTcpSlaveNetwork component:

BooleanValue – Read/Write Boolean values,
NumericValue – Read/Write Numeric values.

WARNING! There is only one table for booth values. Data points addresses are assigned manually, please take care not to override one register from many components.

WARNING! BooleanValue and NumericValue have read and write function. To read only use only Out slot, In slot live not connected with null (for Boolean) or nan (for numeric) value.

WARNING! Using controller outputs in Sedona application will disable writing faction to controller outputs registers. In this case Sedona application has higher priority.

6.2.1 Boolean Value

Boolean Value component is responsible for reading and writing to controller Modbus table Boolean value. Value can be read only for Modbus Master – Bit Type: Discrete Input or read and write for Modbus Master – Bit Type: Coil.

Writing Boolean value use the same table as numeric values. Addressing component make sure that register is not in use by another component.

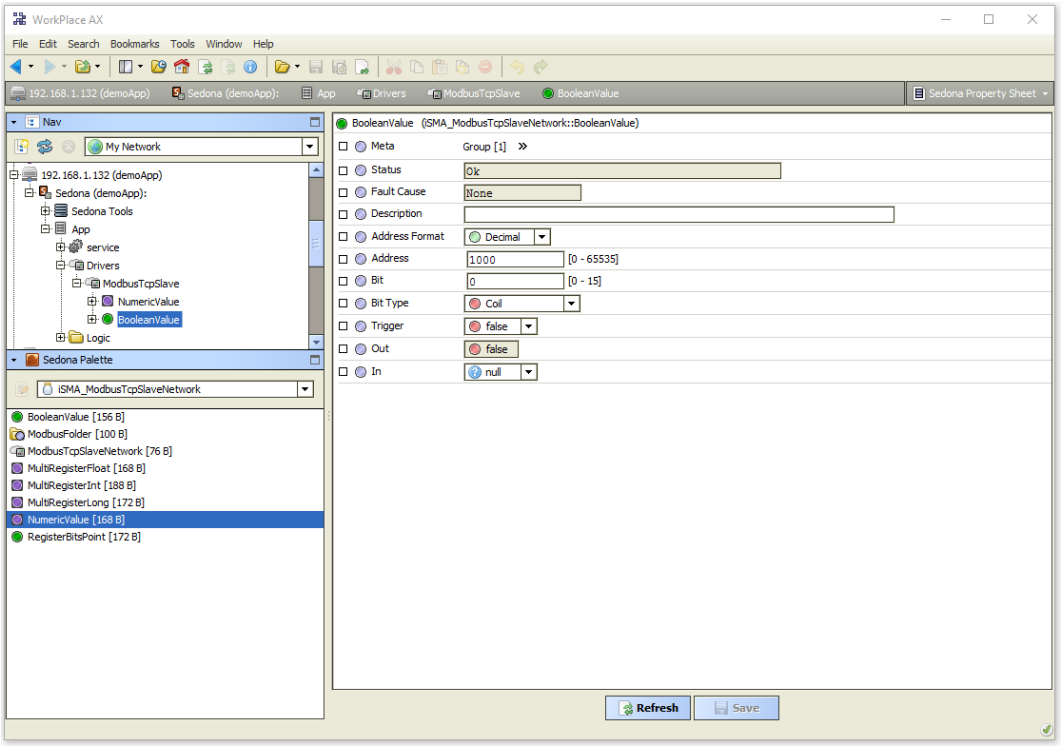


Figure 18 - Modbus TCP Slave Boolean value component

Modbus TCP Slave Boolean Value component has the following slots:
Status – Point status,
Fault Cause – Fault cause description,
Description – Point description label up to 32 characters,
Address Format – Modbus addressing format: Modbus/Decimal

Address – Register address (from0 to 65535),

Bit – Bit number in 16-bits register (from 0 to 15)

Bit Type – Bit type for Master Device: Coil – read/write, Discrete Input – read only,

Trigger – Remote force Input value writing to controller Modbus table trigger (on rising edge),

Out – Output slot, current value of device registry,

In – Input slot.

6.2.2 Numeric Value

Numeric Value component is responsible for reading and writing to controller Modbus table numeric value. Value can be read only for Modbus Master – Register Type: Input Register or read and write for Modbus Master – Register Type: Holding Register.

Writing Boolean value use the same table as numeric values. Addressing component make sure that register is not in use by another component.

WARNING! Data Type: Long, SLong, Float use 32-bits format and use two registers. Next free register in table is Register address + 2.
For example: Float value register has 1010 address, next register must have 1012 address.

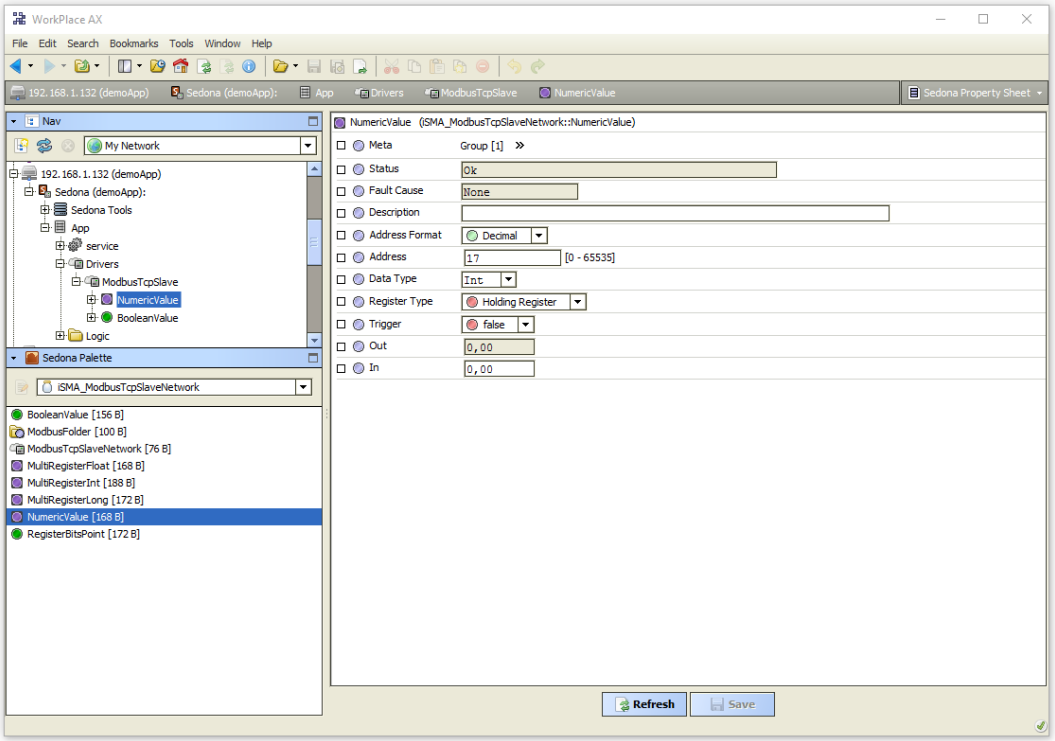


Figure 19 - Modbus TCP Slave Numeric value component

Modbus TCP Slave Numeric Value component has the following slots:

Status – Point status,

Fault Cause – Fault cause description,

Description – Point description label up to 32 characters,

Address Format – Modbus addressing format: Modbus/Decimal

Address – Register address (from0 to 65535),

Data Type – Variable data type: Int, Sint, Long, Slong, Float
WARNING! Long, Slong and Float are 32bit and they use 2 registers

Register Type – Register type for Master Device: Holding Register – read/write, Input Register – read only,

Trigger – Remote force Input value writing to controller Modbus table trigger (on rising edge),

Out – Output slot, current value of device registry,

In – Input slot.

6.3 Modbus Folder

ModbusFolder is a component which groups and organizes Modbus points components. Because ofSedona components name are limited to 7 characters, ModbusFolder has Description Slot where we can us up to 32 characters.

6.4 Modbus RS 485 Slave Device (COM1)

iSMA-B- AAC20 controller can be also Slave Device working on RS485 bus connection. To connect iSMA-B-AAC20 controller to RS485 bus you must use Com1 port with RJ12 connector. RJ12 connector is located between SD card and USB slot. The Modbus RS485 Slave use the same Modbus table as Modbus TCP Slave. From addresses from 0 – 999 (decimal) you can read controller specified data (see Modbus iSMA-B-AAC20 table chapter) and at 1000 – 2999 address range you can add user values by placing data components under ModbusTcpSlaveNetwork (see Modbus TCP Data points chapter).

WARNING! This port is always enable and is ready to use after power up. It can be also used by Sedona Updater software (action, file sending) without turning device to bootloader mode. It also cooperate with iSMA devices (MINI and MIX IP series) Modbus IP/RS485 gateway function. From software side controller is shown as an IP device.

COM1 default settings:

Modbus Address – 1 (register number – 2 Dec.),
Baud Rate – 115200 (value 11520, register number – 135 Dec.),
Stop Bits – 1 (register number – 136 Dec.),
Data Bits – 8 (register number – 137 Dec.),
Parity bits – None (register number – 138 Dec.),
Response Delay – 0 (register number – 139 Dec.),
Modbus Protocol Type - RTU (value 0, register number – 141 Dec.).

6.4.1 Modbus Slave COM1 wiring

WARNING! This port can only work in slave mode and it is not possible to run it in master mode.

To read controller Modbus table by RS485 bus it is necessary to use COM1 port. The port is implemented by RJ12 connector located between SD card and USB slot. The connector provides Modbus bus wires, ground potential G0, and power supply directly connected to G terminal from power supply connector (this option will be used in future to supply Sedona panels). Wiring diagram is shown in Figure bellow.

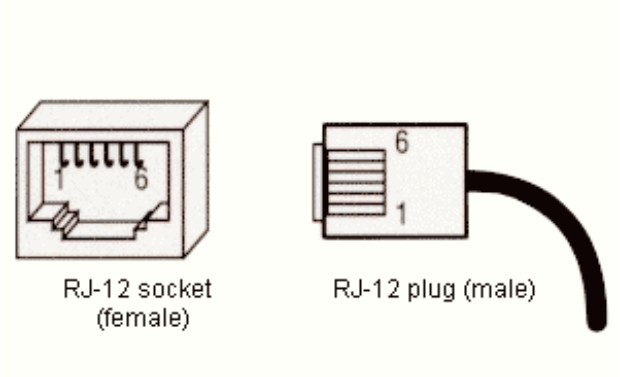


Figure 20 - RJ12 connector pins numbers

RJ12 pins description:

Pin1 – G0 potential, (SD card side),
Pin2 – RS485 – (B),
Pin3 – RS485 + (A),
Pin4 – G0 potential,
Pin5 – G potential, directly connecter to G terminal in power supply,
Pin6 - G potential, directly connecter to G terminal in power supply (USB side).

7 Gateway mode

The iSMA-B-AAC20 controller can also work as a Modbus TCP/RS485 gateway. Default, this option is enable until there is no ModbusAsyncNetwork component in Sedona application or the component is disable (ModbusAsyncNetwork -> Enable slot in false state).

RS485 communication parameters can be setup by:

Controller configuration Web Page,
Modbus Async Network component (remember to set false in Enable slot),
Controller Modbus register table (see iSMA-B-AAC20 Modbus table chapter).

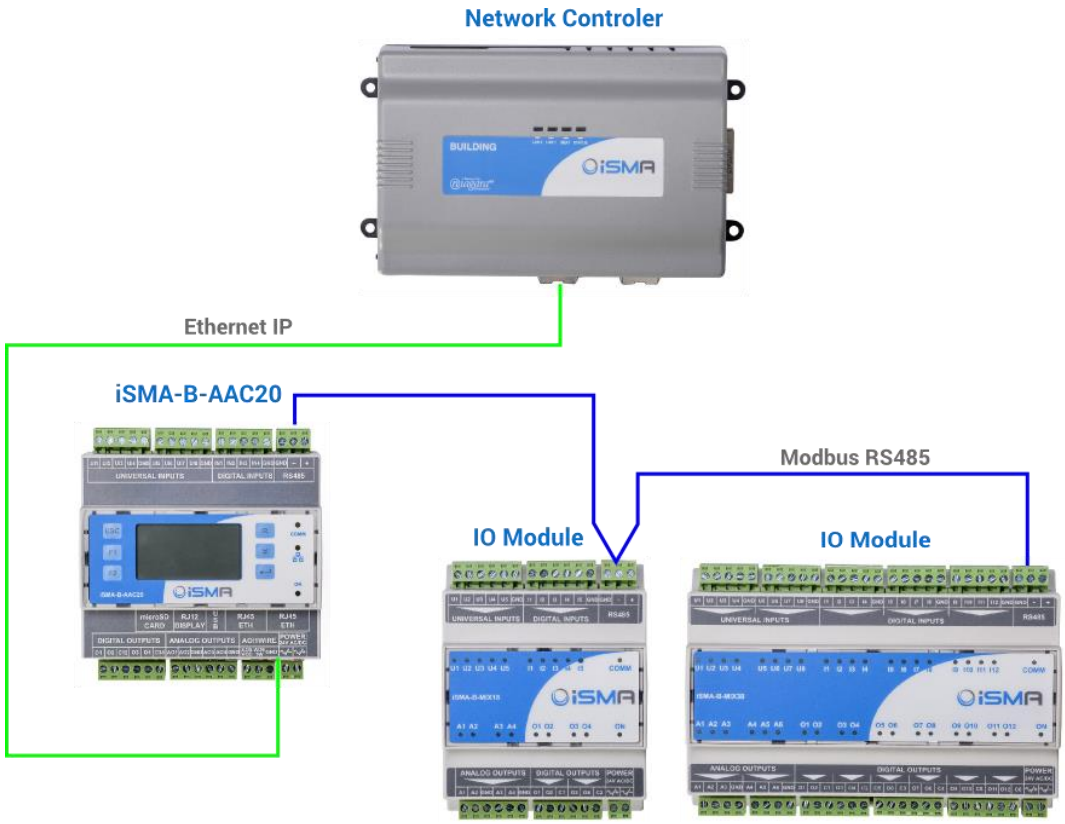


Figure 21 - Modbus TCP gateway topology

8 iSMA Module

iSMA Modules are a Modbus Async Network extension to serve iSMA devises from MIX, MINI and wireless series using Modbus protocol. iSMA Modules kit contains prepared components for serving physical inputs, outputs and configuration parameters. iSMA Modules are an extension of Modbus Async Network designed to easily serve iSMA devices series like MIX modules, MINI modules and Wireless Module using Modbus ASCII/RTU protocol. iSMA Modules kit consists of 4 types of components:

Modbus Network ,
iSMA Device,
iSMA IO Points,
iSMA device config,
iSMA points folder.

WARNING! iSMA Device must be placed under Modbus Network component from ModbusAsyncNetwork kit.

8.1 iSMA Device

iSMA Device is a component designed to cooperate with iSMA Devices hardware. This component has built-in parameters to work with all iSMA Device in ModbusAsyncNetwork.

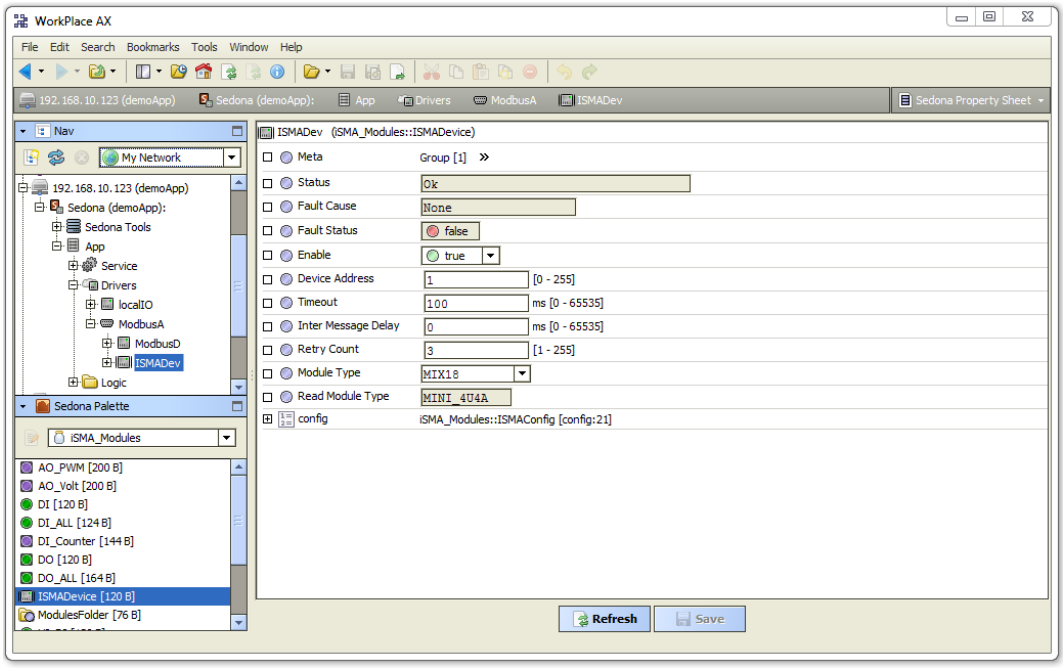


Figure 22 – iSMA Module Device component

8.2 iSMA Device config component

iSMA config is a special component dedicated to set up iSMA series device parameters. Adding and removing config component is done by Module Type slot in iSMA Device component. To add configuration component you must chose proper module type from drop-down list. If the connection is established module type is displayed in Read Module Type slot.

WARNING! This component has not auto refresh option. To read or to write device configuration you must use component action. It is recommended to read module configuration before changing parameters.

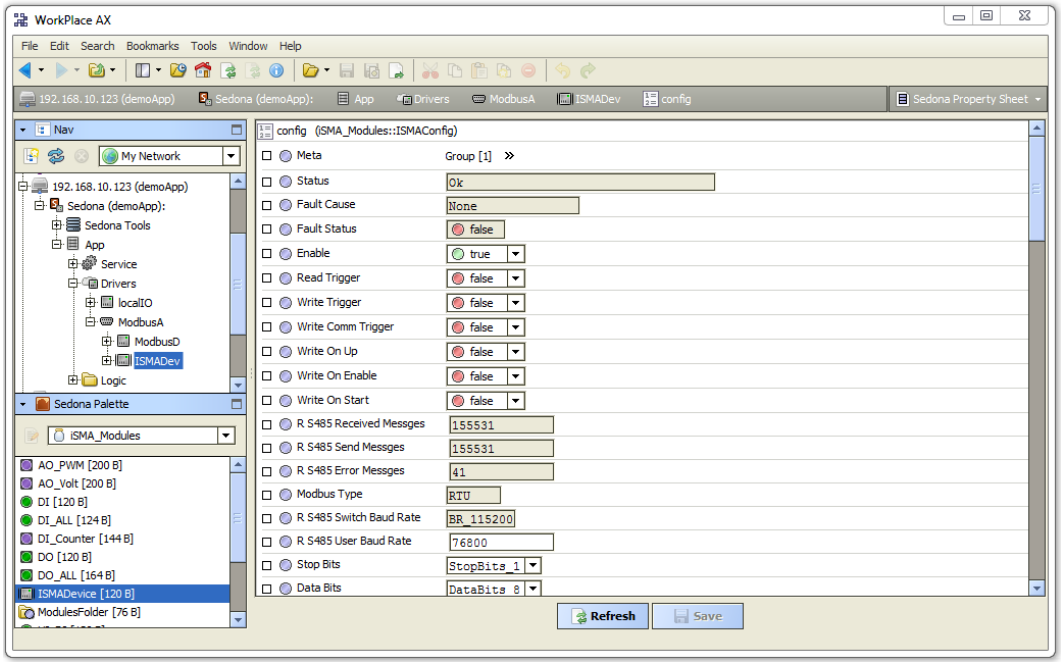


Figure 23 – iSMA Module config component

8.3 iSMA IO components

iSMA Module was created to simplify commission process for iSMA series device MIX, MINI and Wireless series. iSMA Module kit contains of components to serve all input/output.

8.3.1 DI components

In iSMA Module kit are available two types of component to read device Digital Inputs: DI – to read individual Digital Inputs (input number is selected in component property sheet), DI_ALL to read all Digital Inputs using one register.

Note: DI_ALL has 12 inputs slots (DI up to 12 inputs) which corresponds to the largest module iSMA-B-MIX38. Using module with smaller input number causes that inputs above module inputs number are inactive and always in false state.

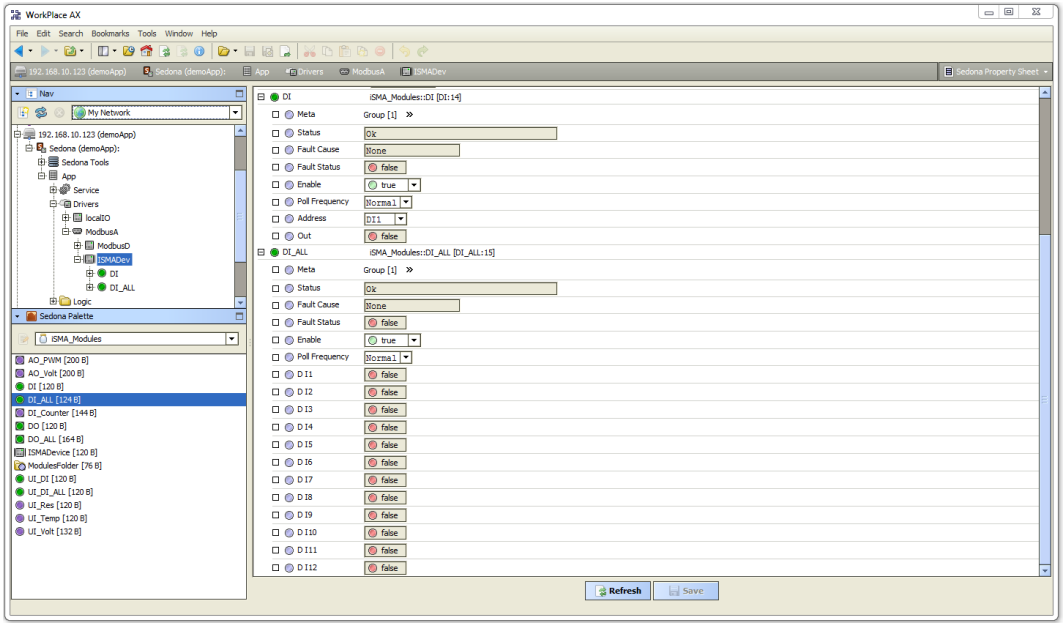


Figure 24 – iSMA Module Digital Input components

8.3.2 DO components

In iSMA Module kit are available two types of component to read/write device Digital Outputs: DO – to read/write individual Digital Outputs (output number is selected in component property sheet), DO_ALL to read all Digital Outputs using one register.

Note: DO_ALL has 12 inputs slots (DO up to 12 outputs) which corresponds to the largest module iSMA-B-MIX38. Using module with smaller output number causes that outputs above module outputs number are inactive.

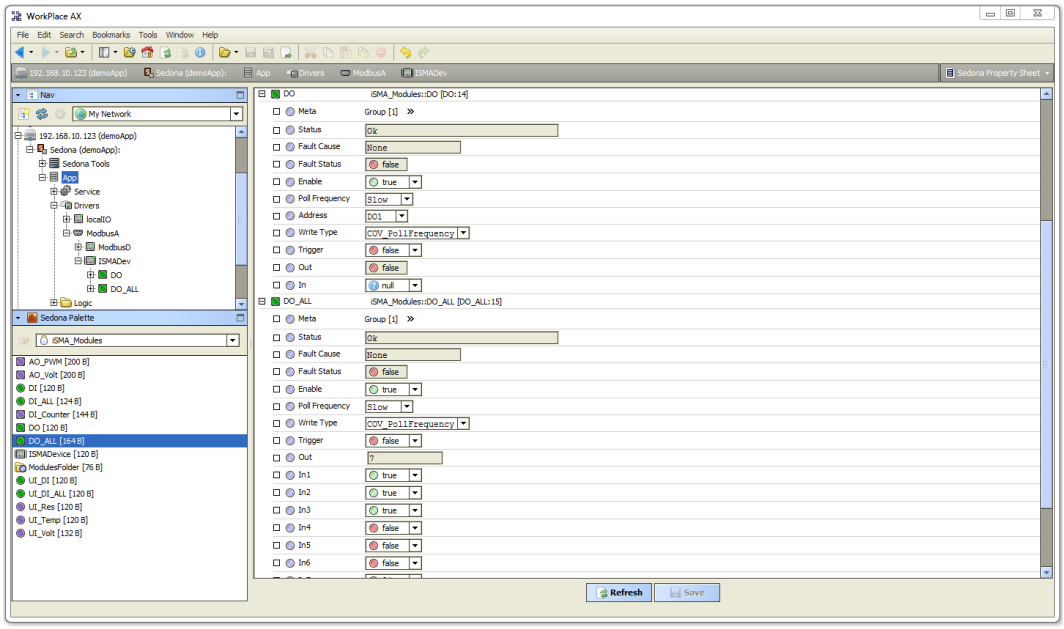


Figure 25 - iSMA Module Digital Output components

8.3.3 UI components

In iSMA Module kit are available five types of component to read device Universal Inputs:

UI_Temp – to read temperature value from NTC sensor connected to Input,
UI_Res - to read resistance value between Universal Input and G0,
UI_Volt – to read voltage value between Universal Input and G0,
UI_DI – to read Boolean value (dry contact) from single Universal Input,
UI_DI_ALL - to read Boolean value (dry contact) from all Universal Inputs in one register.

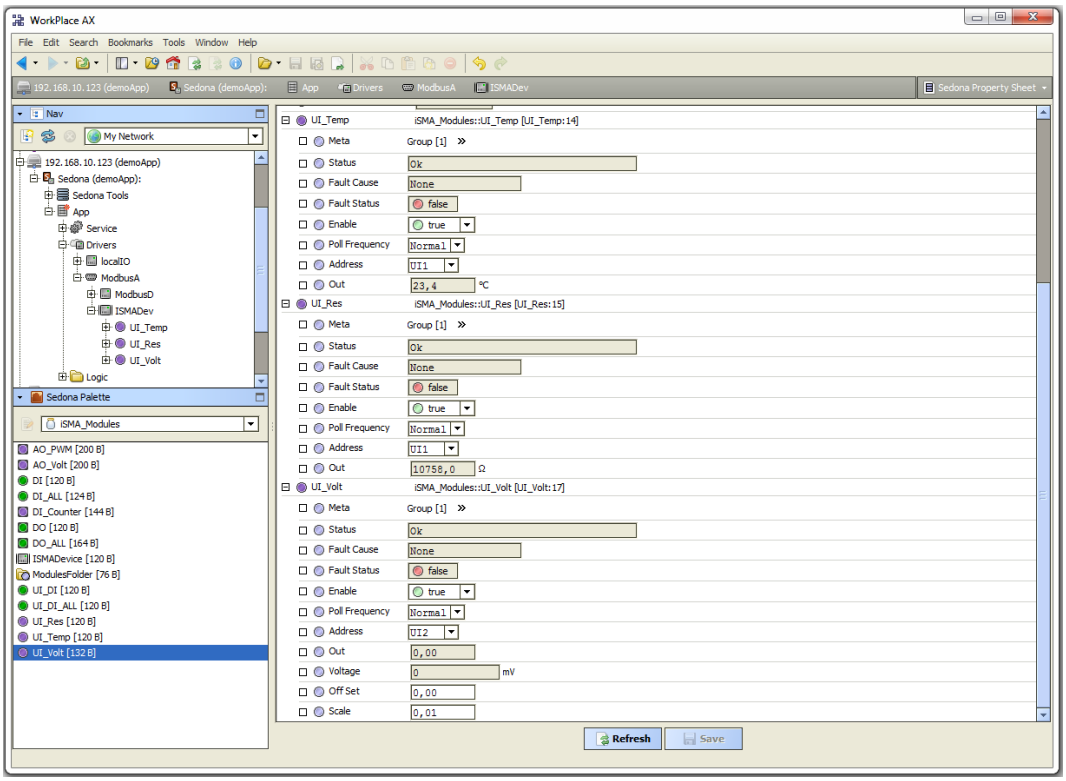


Figure 26 - iSMA Module Universal Input components

Universal Input configuration (sensor type, voltage measurement, filter time and resolution) is done in module config component.

Input number is selected in component property sheet, sensor type in module config component.

Note: UI components have 8 inputs, which corresponds to the largest module iSMA-B-MIX38 and Mini iSMA-B-8U. Using module with smaller input number causes that inputs above module inputs number are inactive and always have 0 value.

8.3.4 AO components

In iSMA Module kit are available two types of component to read/write device Analog Outputs:

AO_Volt – to set up voltage signal (0 – 10000mV) on Analog Output,
AO_PWM – to set up PWM signal (0 – 100%).

Selection if output works in voltage or PWM mode is made in module config component.

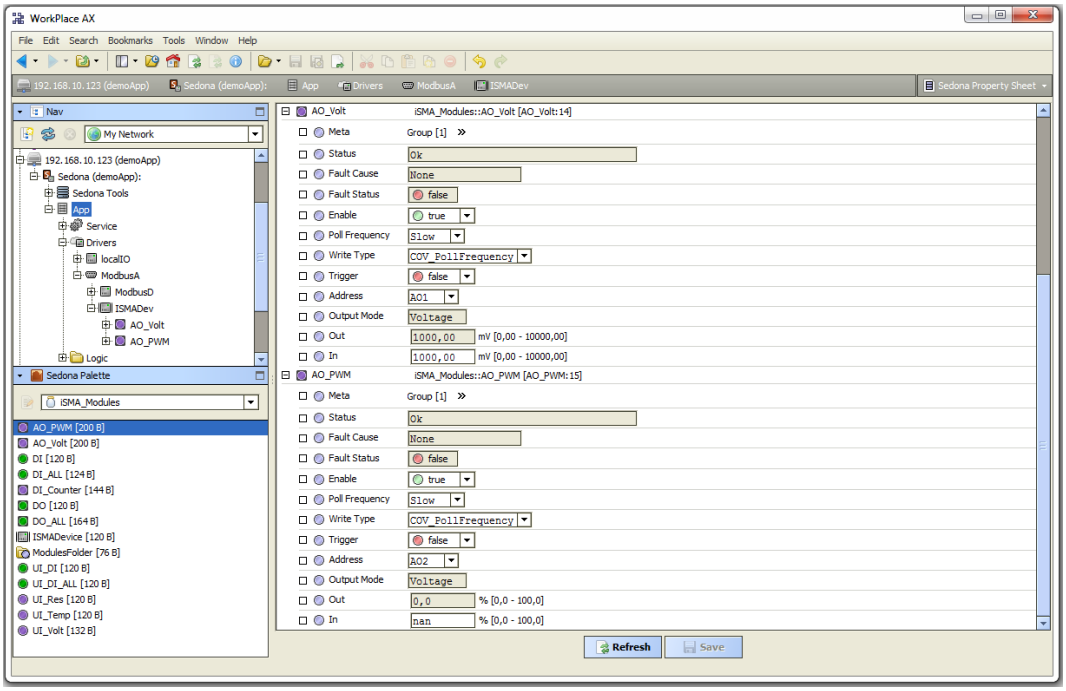


Figure 27 - Analog Outputs property sheet view

Input number is selected in component property sheet, sensor type in module config component. Notice that AO components have 6 inputs, which corresponds to the largest module iSMA-B-MIX38. Using module with smaller output number causes that outputs above module outputs number are inactive.

8.3.5 iSMA Folder

iSMAFolder is a component which groups and organizes iSMA Module IO points components. Because of Sedona components name are limited to 7 characters, ModbusFolder has Description Slot where we can use up to 32 characters.

9 iSMA-B-AAC20 Modbus table

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description
40001	0	0x00	VERSION AND TYPE	Read Only	Controller firmware version and type
40002	1	0x01	HARDWARE VERSION	Read Only	Controller Hardware version
40003	2	0x02	MODBUS ADDRESS	Read &Write Memory	Controller Modbus TCP/IP Slave and Modbus RS485 Slave (COM1) address
40016	15	0x0F	STATE OF DIGITAL INPUTS	Read Only	State of Digital Inputs
40017	16	0x10	STATE OF UNIVERSAL INPUTS WORKING AS DIGITAL INPUTS	Read Only	Status of Universal Inputs working as Digital Inputs
40018	17	0x11	STATE OF DIGITAL OUTPUTS	Read & Write Memory	State of Digital Outputs
40019	18	0x12	STATE OF ANALOG OUTPUTS WORKING AS DIGITAL OUTPUTS	Read & Write Memory	State of Analog Outputs working as Digital Outputs
40022	21	0x15	COUNTER RESET	Read & Write Memory	Set bit in register to reset corresponding counter.
40023	22	0x16	COUNTER 1 LSB	Read & Write Memory	32-bit counters for each Digital Input counting pulses.
40024	23	0x17	COUNTER 1 MSB		
40025	24	0x18	COUNTER 2 LSB	Read & Write Memory	
40026	25	0x19	COUNTER 2 MSB		
40027	26	0x1A	COUNTER 3 LSB	Read & Write Memory	
40028	27	0x1B	COUNTER 3 MSB		
40029	28	0x1C	COUNTER 4 LSB	Read & Write Memory	
40030	29	0x1D	COUNTER 4 MSB		
40046	45	0x2D	COUNTER 12 MSB		
30071	70	0x46	UNIVERSAL INPUT VOLTAGE 1	Read Only	Voltage measurement value is expressed in mV. Formula for the current measurements: $I = \frac{U}{500}$ where: U – register value, 500 – value of attached resistor Temperature is expressed in Celsius degrees * 10
30072	71	0x47	UNIVERSAL INPUT TEMPERATURE 1	Read Only	
30073	72	0x48	UNIVERSAL INPUT VOLTAGE 2	Read Only	
30074	73	0x49	UNIVERSAL INPUT TEMPERATURE 2	Read Only	
30075	74	0x4A	UNIVERSAL INPUT VOLTAGE 3	Read Only	
30076	75	0x4B	UNIVERSAL INPUT TEMPERATURE 3	Read Only	
30077	76	0x4C	UNIVERSAL INPUT VOLTAGE 4	Read Only	
30078	77	0x4D	UNIVERSAL INPUT TEMPERATURE 4	Read Only	
30079	78	0x4E	UNIVERSAL INPUT VOLTAGE 5	Read Only	
30080	79	0x4F	UNIVERSAL INPUT TEMPERATURE 5	Read Only	

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description
30081	80	0x50	UNIVERSAL INPUT VOLTAGE 6	Read Only	For a result, divide the registry value by 10: $T = \frac{registervalue}{10}$ Selection of the type sensor is done using UNIVERSAL INPUT CONFIGURATION register from 40151 to 40158 for each input separately
30082	81	0x51	UNIVERSAL INPUT TEMPERATURE 6	Read Only	
30083	82	0x52	UNIVERSAL INPUT VOLTAGE 7	Read Only	
30084	83	0x53	UNIVERSAL INPUT TEMPERATURE 7	Read Only	
30085	84	0x54	UNIVERSAL INPUT VOLTAGE 8	Read Only	
30086	85	0x55	UNIVERSAL INPUT TEMPERATURE 8	Read Only	
30087	86	0x56	UNIVERSAL INPUT VOLTAGE 1	Read Only	
30088	87	0x57	UNIVERSAL INPUT VOLTAGE 2	Read Only	
30089	88	0x58	UNIVERSAL INPUT VOLTAGE 3	Read Only	
30090	89	0x59	UNIVERSAL INPUT VOLTAGE 4	Read Only	
30091	90	0x5A	UNIVERSAL INPUT VOLTAGE 5	Read Only	
30092	91	0x5B	UNIVERSAL INPUT VOLTAGE 6	Read Only	
30093	92	0x5C	UNIVERSAL INPUT VOLTAGE 7	Read Only	
30094	93	0x5D	UNIVERSAL INPUT VOLTAGE 8	Read Only	
30095	94	0x5E	UNIVERSAL INPUT TEMPERATURE 1	Read Only	
30096	95	0x5F	UNIVERSAL INPUT TEMPERATURE 2	Read Only	
30097	96	0x60	UNIVERSAL INPUT TEMPERATURE 3	Read Only	
30098	97	0x61	UNIVERSAL INPUT TEMPERATURE 4	Read Only	
30099	98	0x62	UNIVERSAL INPUT TEMPERATURE 5	Read Only	
30100	99	0x63	UNIVERSAL INPUT TEMPERATURE 6	Read Only	
30101	100	0x64	UNIVERSAL INPUT TEMPERATURE 7	Read Only	
30102	101	0x65	UNIVERSAL INPUT TEMPERATURE 8	Read Only	
30103	102	0x66	RESISTIVE INPUT 1 LSB	Read Only	Resistance measurement result expressed in Ω Value range from 0 Ω to 1 000 000 Ω **Note:** In PT1000 or NI1000 input working type the reading accuracy increase and the register value is multiply by 10
30104	103	0x67	RESISTIVE INPUT 1 MSB	Read Only	
30105	104	0x68	RESISTIVE INPUT 2 LSB	Read Only	
30106	105	0x69	RESISTIVE INPUT 2 MSB	Read Only	
30107	106	0x6A	RESISTIVE INPUT 3 LSB	Read Only	
30108	107	0x6B	RESISTIVE INPUT 3 MSB	Read Only	
30109	108	0x6C	RESISTIVE INPUT 4 LSB	Read Only	
30110	109	0x6D	RESISTIVE INPUT 4 MSB	Read Only	
30111	110	0x6E	RESISTIVE INPUT 5 LSB	Read Only	
30112	111	0x6F	RESISTIVE INPUT 5 MSB	Read Only	

<table>
<tr><th>Modbus Address</th><th>Dec Addr</th><th>Hex Adr.</th><th>Register name</th><th>Access</th><th>Description</th></tr>
<tr><td>30113</td><td>112</td><td>0x70</td><td>RESISTIVE INPUT 6 LSB</td><td>Read Only</td><td rowspan="6"></td></tr>
<tr><td>30114</td><td>113</td><td>0x71</td><td>RESISTIVE INPUT 6 MSB</td><td>Read Only</td></tr>
<tr><td>30115</td><td>114</td><td>0x72</td><td>RESISTIVE INPUT 7 LSB</td><td>Read Only</td></tr>
<tr><td>30116</td><td>115</td><td>0x73</td><td>RESISTIVE INPUT 7 MSB</td><td>Read Only</td></tr>
<tr><td>30117</td><td>116</td><td>0x74</td><td>RESISTIVE INPUT 8 LSB</td><td>Read Only</td></tr>
<tr><td>30118</td><td>117</td><td>0x75</td><td>RESISTIVE INPUT 8 MSB</td><td>Read Only</td></tr>
<tr><td>40121</td><td>120</td><td>0x78</td><td>VALUE OF ANALOG OUTPUT 1</td><td>Read & Write</td><td rowspan="6">The voltage at the Analog Outputs are given in the mV range from 0 to 10000 mV</td></tr>
<tr><td>40122</td><td>121</td><td>0x79</td><td>VALUE OF ANALOG OUTPUT 2</td><td>Read & Write</td></tr>
<tr><td>40123</td><td>122</td><td>0x7A</td><td>VALUE OF ANALOG OUTPUT 3</td><td>Read & Write</td></tr>
<tr><td>40124</td><td>123</td><td>0x7B</td><td>VALUE OF ANALOG OUTPUT 4</td><td>Read & Write</td></tr>
<tr><td>40125</td><td>124</td><td>0x7C</td><td>VALUE OF ANALOG OUTPUT 5</td><td>Read & Write</td></tr>
<tr><td>40126</td><td>125</td><td>0x7D</td><td>VALUE OF ANALOG OUTPUT 6</td><td>Read & Write</td></tr>
<tr><td>40136</td><td>135</td><td>0x87</td><td>RS 485 (COM1) BAUD RATE</td><td>Read & Write Memory</td><td>Transmission baud rate is defined by the user calculated using the formula:
$Baudrate = (registervalue) \cdot 10$
The default value is 11520 (115200 bps)</td></tr>
<tr><td>40137</td><td>136</td><td>0x88</td><td>RS 485 (COM1) STOP BITS</td><td>Read & Write Memory</td><td>Supported values are 1 and 2
The default value 1</td></tr>
<tr><td>40138</td><td>137</td><td>0x89</td><td>RS 485 (COM1) DATA BITS</td><td>Read & Write Memory</td><td>Supported values are 7 and 8
The default value 7</td></tr>
<tr><td>40139</td><td>138</td><td>0x8A</td><td>RS 485 (COM1) PARITY BIT</td><td>Read & Write Memory</td><td>The default value is 0 (no parity)
Allowed values:<table><tr><th>Value</th><th>Description</th></tr><tr><td>0 (default)</td><td>none</td></tr><tr><td>1</td><td>Odd</td></tr><tr><td>2</td><td>Even</td></tr><tr><td>3</td><td>Always 1</td></tr><tr><td>4</td><td>Always 0</td></tr></table></td></tr>
<tr><td>40140</td><td>139</td><td>0x8B</td><td>RS 485 (COM1) RESPONSE DELAY</td><td>Read & Write Memory</td><td>Delay in ms before sending response
The default value is 0.</td></tr>
<tr><td>40141</td><td>140</td><td>0x8C</td><td>WATCHDOG TIME</td><td>Read & Write Memory</td><td>Time in second before watchdog reset in case no transmission.
A value of 0 disables Watchodog.
The default value is 15s</td></tr>
<tr><td>40142</td><td>141</td><td>0x8D</td><td>RS 485 (COM1)
MODBUS PROTOCOL TYPE</td><td>Read & Write Memory</td><td>Protocol Type (40142)
0 – RTU, 1 – ASCII
The default value is 0 - RTU.</td></tr>
</table>

<table>
<tr><th>Modbus Address</th><th>Dec Addr</th><th>Hex Adr.</th><th>Register name</th><th>Access</th><th>Description</th></tr>
<tr><td>40143</td><td>142</td><td>0x8E</td><td>DEFAULT STATE OF DIGITAL OUTPUTS</td><td>Read & Write Memory</td><td>State of Digital Outputs assigned at the start of the module and watchdog reset.
The default value is 0.</td></tr>
<tr><td>40144</td><td>143</td><td>0x8F</td><td>DEFAULT STATE OF ANALOG OUTPUTS (DIGITAL)</td><td>Read & Write Memory</td><td>State of Analog Outputs assigned at the start of the module and watchdog reset.
The default value is 0.</td></tr>
<tr><td>40145</td><td>144</td><td>0x90</td><td>DEFAULT STATE OF ANALOG OUTPUT 1</td><td>Read & Write Memory</td><td rowspan="6">In the registers is stored value in mV of voltage that appears at the Analog Output after power on or watchdog reset.
The default value is 0.</td></tr>
<tr><td>40146</td><td>145</td><td>0x91</td><td>DEFAULT STATE OF ANALOG OUTPUT 2</td><td>Read & Write Memory</td></tr>
<tr><td>40147</td><td>146</td><td>0x92</td><td>DEFAULT STATE OF ANALOG OUTPUT 3</td><td>Read & Write Memory</td></tr>
<tr><td>40148</td><td>147</td><td>0x93</td><td>DEFAULT STATE OF ANALOG OUTPUT 4</td><td>Read & Write Memory</td></tr>
<tr><td>40149</td><td>148</td><td>0x94</td><td>DEFAULT STATE OF ANALOG OUTPUT 5</td><td>Read & Write Memory</td></tr>
<tr><td>40150</td><td>149</td><td>0x95</td><td>DEFAULT STATE OF ANALOG OUTPUT 6</td><td>Read & Write Memory</td></tr>
<tr><td>40151</td><td>150</td><td>0x96</td><td>UNIVERSAL INPUT 1 CONFIGURATION</td><td>Read & Write Memory</td><td rowspan="7">Configuration of Universal Input and type of temperature sensor.
The default value is 1.
<table><tr><th>Value</th><th>Description / Sensor</th></tr><tr><td>0</td><td>Resistance measurement off</td></tr><tr><td>1</td><td>10K3A1 NTC</td></tr><tr><td>2</td><td>10K4A1 NTC</td></tr><tr><td>3</td><td>10K NTC Carel</td></tr><tr><td>4</td><td>20K6A1 NTC</td></tr><tr><td>5</td><td>2,2K3A1 NTC B=3975K</td></tr><tr><td>6</td><td>3K3A1 NTC</td></tr><tr><td>7</td><td>30K6A1 NTC</td></tr><tr><td>8</td><td>SIE1</td></tr><tr><td>9</td><td>TAC1</td></tr></table></td></tr>
<tr><td>40152</td><td>151</td><td>0x97</td><td>UNIVERSAL INPUT 2 CONFIGURATION</td><td>Read & Write Memory</td></tr>
<tr><td>40153</td><td>152</td><td>0x98</td><td>UNIVERSAL INPUT 3 CONFIGURATION</td><td>Read & Write Memory</td></tr>
<tr><td>40154</td><td>153</td><td>0x99</td><td>UNIVERSAL INPUT 4 CONFIGURATION</td><td>Read & Write Memory</td></tr>
<tr><td>40155</td><td>154</td><td>0x9A</td><td>UNIVERSAL INPUT 5 CONFIGURATION</td><td>Read & Write Memory</td></tr>
<tr><td>40156</td><td>155</td><td>0x9B</td><td>UNIVERSAL INPUT 6 CONFIGURATION</td><td>Read & Write Memory</td></tr>
<tr><td>40157</td><td>156</td><td>0x9C</td><td>UNIVERSAL INPUT 7 CONFIGURATION</td><td>Read & Write Memory</td></tr>
</table>

<table>
<tr><th>Modbus Address</th><th>Dec Addr</th><th>Hex Adr.</th><th>Register name</th><th>Access</th><th>Description</th></tr>
<tr><td>40158</td><td>157</td><td>0x9D</td><td>UNIVERSAL INPUT 8 CONFIGURATION</td><td>Read & Write Memory</td><td><table><tr><td>10</td><td>SAT1</td></tr><tr><td>16</td><td>Pt1000</td></tr><tr><td>17</td><td>Ni1000</td></tr><tr><td>+128</td><td>Voltage measurement off</td></tr></table></td></tr>
<tr><td>40159</td><td>158</td><td>0x9E</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 1</td><td>Read & Write Memory</td><td rowspan="8">Filter time constant, expressed in seconds in the range from 0 to 60 seconds.

A value of 0 disables the filter.

The default value is 2s.</td></tr>
<tr><td>40160</td><td>159</td><td>0x9F</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 2</td><td>Read & Write Memory</td></tr>
<tr><td>40161</td><td>160</td><td>0xA0</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 3</td><td>Read & Write Memory</td></tr>
<tr><td>40162</td><td>161</td><td>0xA1</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 4</td><td>Read & Write Memory</td></tr>
<tr><td>40163</td><td>162</td><td>0xA2</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 5</td><td>Read & Write Memory</td></tr>
<tr><td>40164</td><td>163</td><td>0xA3</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 6</td><td>Read & Write Memory</td></tr>
<tr><td>40165</td><td>164</td><td>0xA4</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 7</td><td>Read & Write Memory</td></tr>
<tr><td>40166</td><td>165</td><td>0xA5</td><td>FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 8</td><td>Read & Write Memory</td></tr>
<tr><td>40167</td><td>166</td><td>0xA6</td><td>RESOLUTION OF THE UNIVERSAL INPUTS</td><td>Read & Write Memory</td><td>Resolution of Universal Inputs. When bit is set measurement at corresponding input is done with 16-bit resolution.
By default, all measurements are done with 12-bit resolution.</td></tr>
<tr><td>40168</td><td>167</td><td>0xA7</td><td>ANALOG OUTPUT 1 CONFIGURATION</td><td>Read & Write Memory</td><td rowspan="4">By default, all measurements are done with 12-bit resolution.

Configuring the mode of Analog Output according to the following table:<table><tr><th>Value</th><th>Description</th></tr><tr><td>0 (default)</td><td>Voltage output 0-10V</td></tr><tr><td>1</td><td>PWM 1Hz</td></tr><tr><td>2</td><td>PWM 10Hz</td></tr><tr><td>3</td><td>PWM 100Hz</td></tr></table></td></tr>
<tr><td>40169</td><td>168</td><td>0xA8</td><td>ANALOG OUTPUT 2 CONFIGURATION</td><td>Read & Write Memory</td></tr>
<tr><td>40170</td><td>169</td><td>0xA9</td><td>ANALOG OUTPUT 3 CONFIGURATION</td><td>Read & Write Memory</td></tr>
<tr><td>40171</td><td>170</td><td>0xAA</td><td>ANALOG OUTPUT 4 CONFIGURATION</td><td>Read & Write Memory</td></tr>
</table>

<table>
<tr><th>Modbus Address</th><th>Dec Addr</th><th>Hex Adr.</th><th>Register name</th><th>Access</th><th>Description</th></tr>
<tr><td>40172</td><td>171</td><td>0xAB</td><td>ANALOG OUTPUT 5 CONFIGURATION</td><td>Read & Write Memory</td><td rowspan="2"><table><tr><td>4</td><td>PWM 0.1Hz</td></tr><tr><td>5</td><td>PWM 0.01Hz</td></tr></table></td></tr>
<tr><td>40173</td><td>172</td><td>0xAC</td><td>ANALOG OUTPUT 6 CONFIGURATION</td><td>Read & Write Memory</td></tr>
</table>