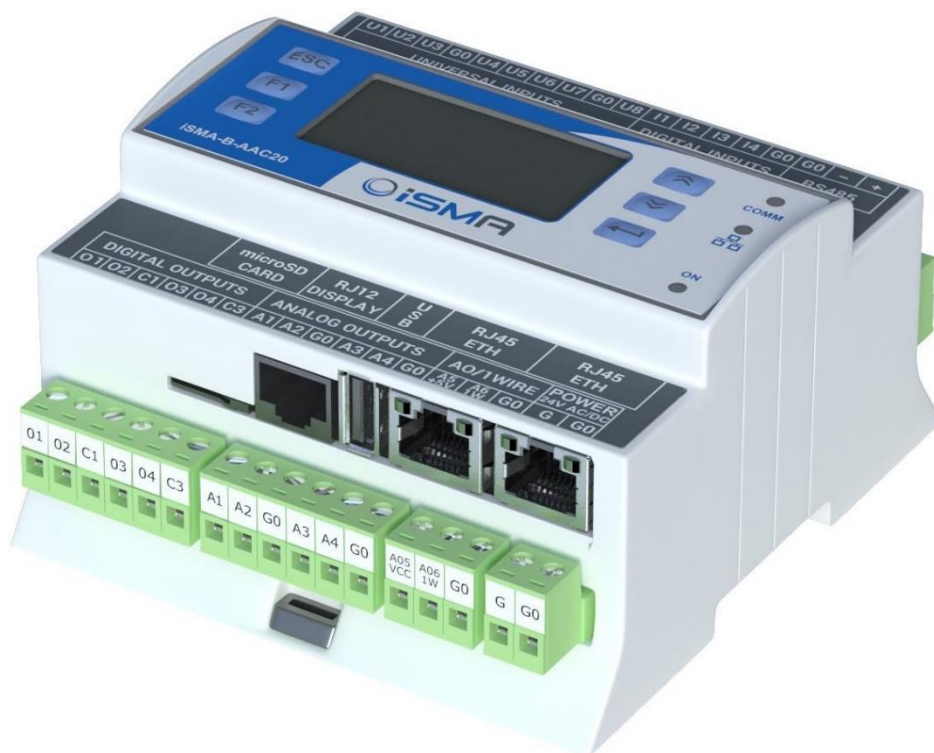


iSMA-B-AAC20

User Manual

Sedona Modbus



Powered by
sedona
FRAMEWORK™

Global Control 5 S.A.
Warsaw, Poland
www.globalcontrol5.com

Table of Contents

1	Introduction	4
1.1	Revision history.....	4
2	Sedona Modbus.....	4
3	Installing iSMA Modbus kits.....	6
4	Modbus Async Network kit.....	7
4.1	Modbus Async license and limitation.....	7
4.2	Modbus Async Network.....	7
4.3	Modbus Async Device	9
4.4	Modbus data points	11
4.4.1	Modbus Async Boolean Point	11
4.4.2	Modbus Async Boolean Writable.....	13
4.4.3	Modbus Async Numeric Point.....	14
4.4.4	Modbus Async Numeric Writable	15
4.4.5	Modbus Async Numeric Multi Point	17
4.5	Modbus folder	18
5	Modbus TCP Network kit	20
5.1	Modbus TCP license and limitation	20
5.2	Modbus TCP Network	20
5.3	Modbus TCP Device.....	22
5.4	Modbus TCP data points	24
5.4.1	Modbus TCP Boolean Point.....	24
5.4.2	Modbus TCP Boolean Writable	26
5.4.3	Modbus TCP Numeric Point	27
5.4.4	Modbus TCP Numeric Writable.....	29
5.4.5	Modbus TCP Numeric Multi Point.....	30
5.5	Modbus folder	32
6	Modbus TCP Slave Network kit	32
6.1	Modbus TCP Slave Network.....	34
6.2	Modbus TCP Slave data points	35
6.2.1	Boolean Value	35
6.2.2	Numeric Value	36

6.3 Modbus Folder	38
7 Modbus RJ12 Network kit	38
7.1 Modbus RJ12 license and limitation	38
7.2 Modbus RJ12 Network.....	39
7.3 Modbus RJ12 Device.....	40
7.4 Modbus data points	42
7.5 Modbus RJ12 wiring.....	42
8 Gateway mode	44
9 iSMA Module	45
9.1 iSMA Device.....	45
9.2 iSMA Device config component	46
9.3 iSMA IO components	47
9.3.1 DI components	47
9.3.2 DO components.....	48
9.3.3 UI components	49
9.3.4 AO components.....	49
9.3.5 iSMA Folder	50
10 iSMA-B-AAC20 Modbus table	51

1 Introduction

This manual contains information about Modbus protocols in the iSMA-B-AAC20 controller. The iSMA-B-AAC20 controller supports the following types of Modbus protocols:

- Modbus ASCII/RTU;
- Modbus TCP;
- Modbus TCP Slave.

1.1 Revision History

Rev	Date	Description
1.0	28.08.2015	First edition
1.3	20.04.2017	<ul style="list-style-type: none"> • 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
1.5	27.01.2020	<ul style="list-style-type: none"> • Added RJ12 network description • Added Safety Rules, Technical Specification and Dimensions • Updated Modbus register table • Replaced environment of programming from Workplace to iSMA Tool • BTL compliance

Table 1. Revision history

2 Sedona Modbus

The Modbus protocol defines a message structure and format used in communication transactions. The 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 the Modbus kits, please check if the latest kit version is used. The latest kits are available in the AAC20 Software Bundle on GC5 support website: www.support.gc5.pl

2.1 Modbus Registers

A Modbus device holds transient (real-time) data and persistent (configuration) data in the addressable registers. Here, the term “registers” implies all addressable data, but this is a loose interpretation. Using a 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; the 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; the Modbus Master can read (only) inputs;
- **Input register:** 16-bit registers that store data collected from the field by the Modbus Slave; the Modbus Master can read (only) input registers;
- **Holding register:** 16-bit registers that store general-purpose data in the Modbus Slave the Modbus Master can both read from and write to input registers.

2.2 Modbus Data Addresses

The 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 5 Modbus kits in the iSMA-B-AAC20 controller:

- **iSMA Modbus Async Network:** To serve the Modbus RS485 Master port of the iSMA-B-AAC20 controller,
- **iSMA Modules:** Extension of the Modbus Async Network to serve the iSMA MIX, Wireless and MINI devices using Modbus Async protocol,
- **iSMA Modbus TCP Network:** To serve the Modbus TCP Master of the iSMA-B-AAC20 device using IP connection,
- **iSMA Modbus TCP Slave Network:** To serve the Modbus TCP Slave of the iSMA-B-AAC20 controller using IP connection,
- **iSMA Modbus RJ12 Network:** To serve the Modbus Async using RJ12 connection.

To install the Modbus kits, import the kits to the iSMA Tool (possibly as part of the package of various kits in a zip file). To do this, choose on the top bar menu Sedona -> Import Sedona Files.

After a successful import of the files, upload the files to your device using the Kit Manager Application.

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

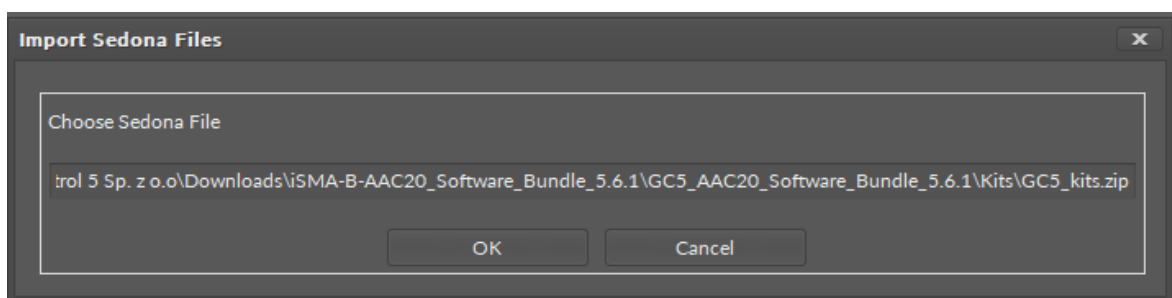


Figure 1. Importing newest Sedona kits to the iSMA Tool

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 the Modbus points. The iSMA-B-AAC20 controller has one RS485 port which can be used as a Modbus RTU / ASCII Master.

The 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 a standard license there are available 500 data points and this number cannot be expanded. The number of available points is shown in the 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 ModbusAsyncNetwork Component

The ModbusAsyncNetwork is the main component which is responsible for servicing the RS485 physical port. The component must be placed under the Drivers folder. The 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-click or in the Object Properties window:

- **Reset Stats:** Resets network's statistics and starts counting from the beginning;
- **Enable/Disable:** On/Off switching of the Modbus Network.

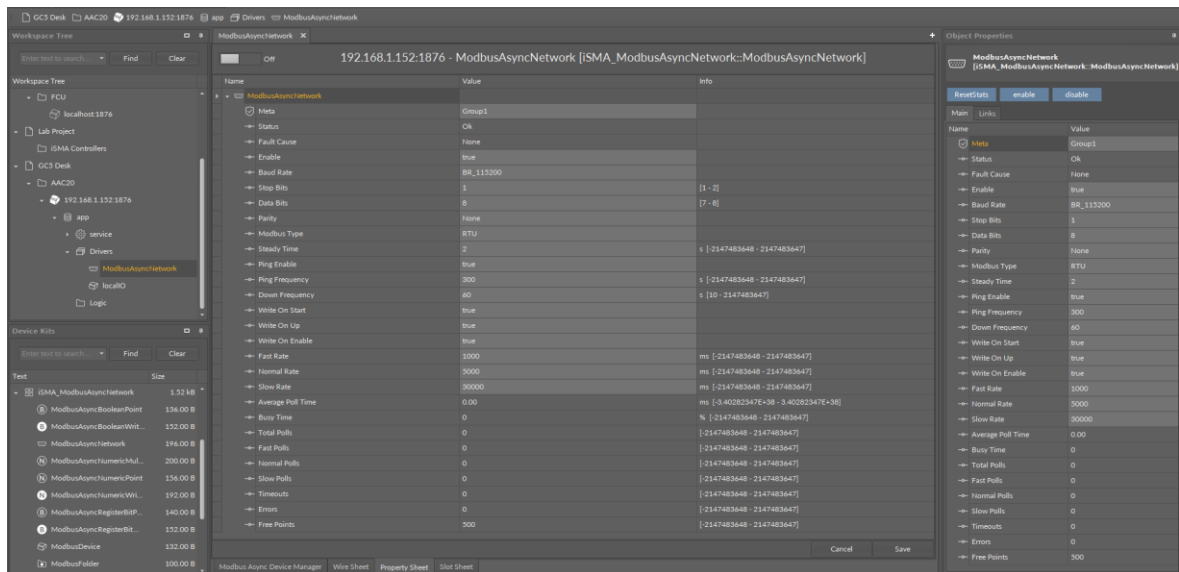


Figure 2. Modbus Async Network component

The ModbusAsyncNetwork component has the following slots:

- **Status:** Network's status, available states:
 - OK: Network is working properly,
 - Disabled: Network is disabled (Slot "Enable" is in false),
 - OK some device/point down: error in the device or points;
- **Fault Cause:** Fault cause description;
- **Enable:** This option switches on or switches off the Modbus Network („true"- Network enabled, „false"- Network disabled),
- **Steady Time:** Network start-up delay time after a power-up or reset;
- **Baud Rate:** The 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:** Enables testing of the device's connection ;
- **Ping Frequency:** Time between testing messages to check the device's connection;
- **Down Frequency:** Time between testing messages for devices or points which have got status down,
- **Write On Start:** Executes a write action in device „Writable" components in the Modbus Network after a reset or power-up;
- **Write On Up:** Executes a write action in device „Writable" components in the Modbus Network after restoring of the connection with Modbus device;
- **Write On Enable:** Executes a write action in device „Writable" components in the Modbus Network after enabling 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 of one message;
- **Busy Time:** Percentage of Modbus Network usage;
- **Total Polls:** Total number of messages;
- **Fast Polls:** Number of messages sent in „Fast“ mode;
- **Normal Polls:** Number of messages sent in „Normal“ mode;
- **Slow Polls:** Number of messages sent in „Slow“ mode;
- **Timeouts:** Number of lost messages, the difference between sent and received messages;
- **Errors:** Number of error messages (for example, with the wrong CRC);
- **Free points:** Number of available physical points in Modbus Network.

4.3 ModbusAsyncDevice Component

The ModbusAsyncDevice is a component which is responsible for servicing a physical device connected to the Modbus Network. The AAC20 device acts as a 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 a Ping action available under the right-click, which sends a test message to the device to check the 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. If any response from the device is received, including an exception response, this is considered a proof of communication, and the Modbus client device is no longer considered “down” if it was previously marked “down”.

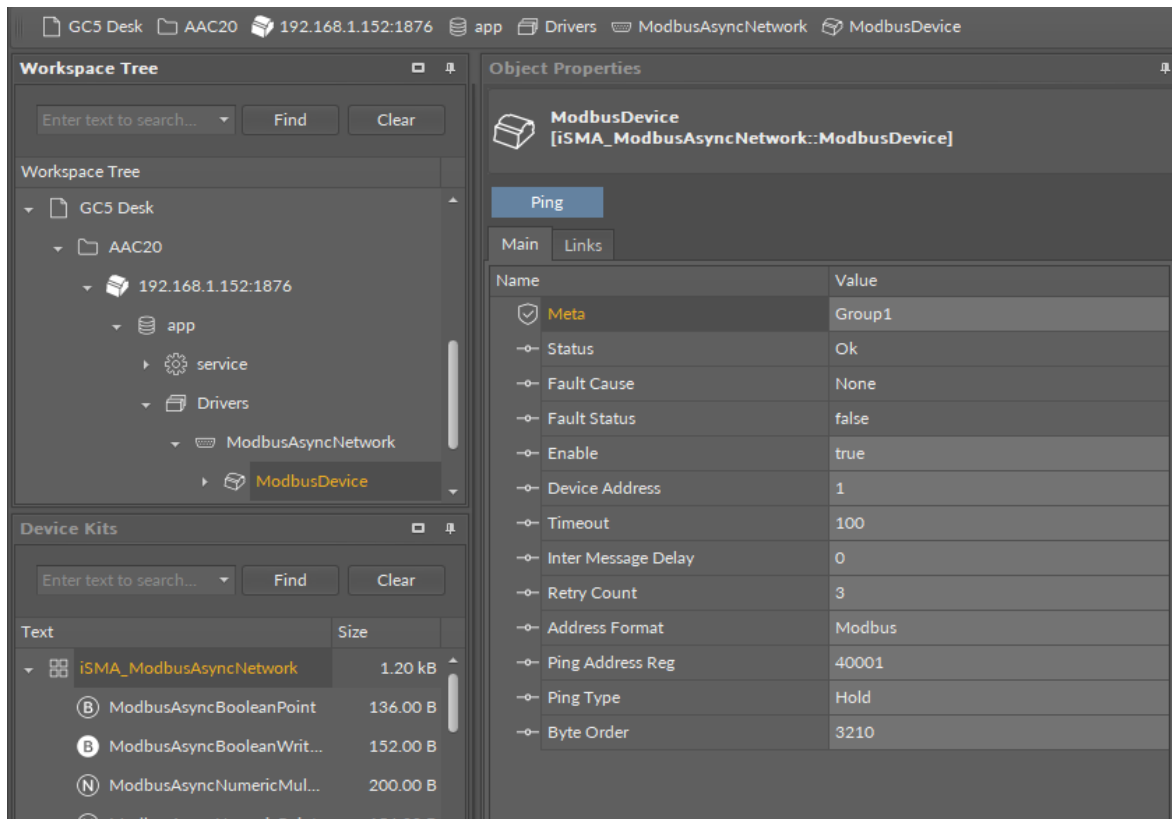


Figure 3. Modbus Async Device component

The ModbusAsyncDevice component has the following slots:

Status: Device's actual status (read-only), can take the following states:

- OK: Device is working properly,
- Disable: Device is disabled (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: Enables/Disables the device;

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

Timeout: Max. device response time from the device request;

Inter Message Delay: Time between messages sent to the device;

Retry Count: Max. number of error messages (CRC error, lost messages);

Address Format: Modbus address format (Modbus, Decimal);

Ping Address Reg: Input or Holding type register's number which will be read for device connection test;

Ping Type: Tested register type: Input/Holding;

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

4.4 Modbus Data Points

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

Boolean Point: Reads Boolean values (Modbus command 0x02);

Boolean Writable: Reads/writes Boolean values (Modbus command 0x05);

Numeric Point: Reads numeric values (Modbus commands: Input - 0x04, Holding - 0x03);

Numeric Writable: Reads/writes numeric values (Modbus commands: 16-bits Int, SInt - 0x06,
32-bits Long, SLong, Float: 0x16);

Numeric Multi Point: Reads up to eight 16-bits registers (Modbus command 0x16).

4.4.1 ModbusAsyncBooleanPoint Component

The ModbusAsyncBooleanPoint is a component which is responsible for reading Boolean values from the device. The component has a Read action available under the right-click, which forces the reading of the point.

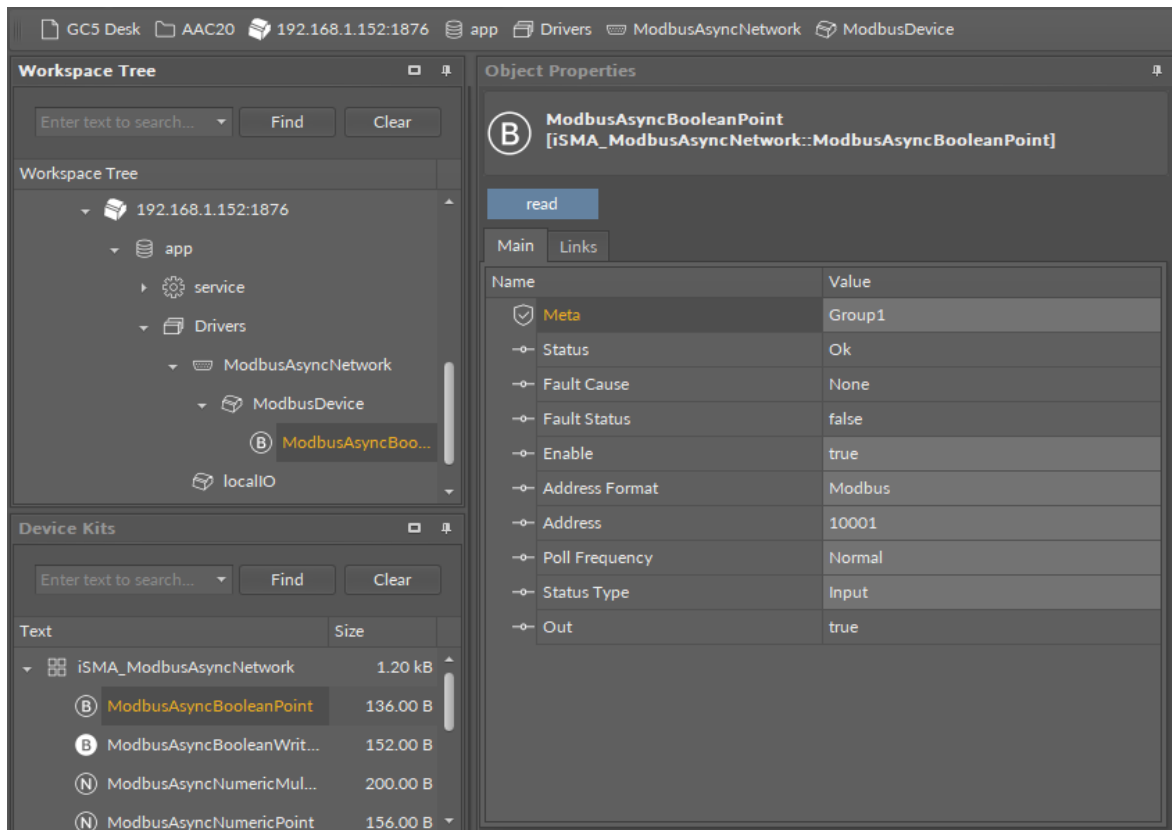


Figure 4. The ModbusAsyncBooleanPoint component

The ModbusAsyncBooleanPoint component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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 the 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:** Enables/Disables the Point (true - Point enabled, false- Point disabled);
- **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 the read register.

4.4.2 ModbusAsyncBooleanWritable Component

The ModbusAsyncBooleanWritable is a component which is responsible for sending and reading Boolean values from the device.

The component has the following actions available under the right-click:

Set True/Set False: Writes a value to the In slot and sends it to the device (not active when slot In has a connected link);

Write: Sends a value from the In slot to the device;

Read: Reads a value from the device and sends to the Out slot.

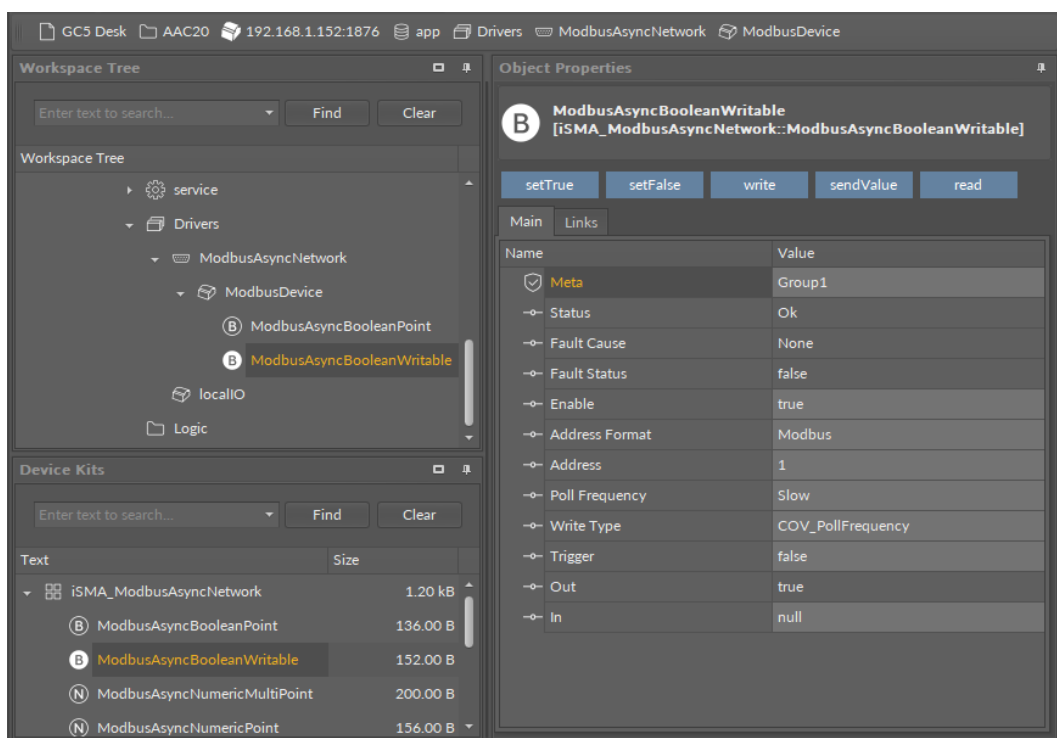


Figure 5. Modbus Async Boolean Writable component

The ModbusAsyncBooleanWritable component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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 the 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:** Enables/Disables the Point (true - Point enabled, false - Point disabled),
- **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 triggered by COV);
- **Trigger:** Forcefully send the value (on rising edge), regardless of the current poll mode;
- **Out:** Output slot, the current value of read/write register;
- **In:** Input slot.

4.4.3 ModbusAsyncNumericPoint Component

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

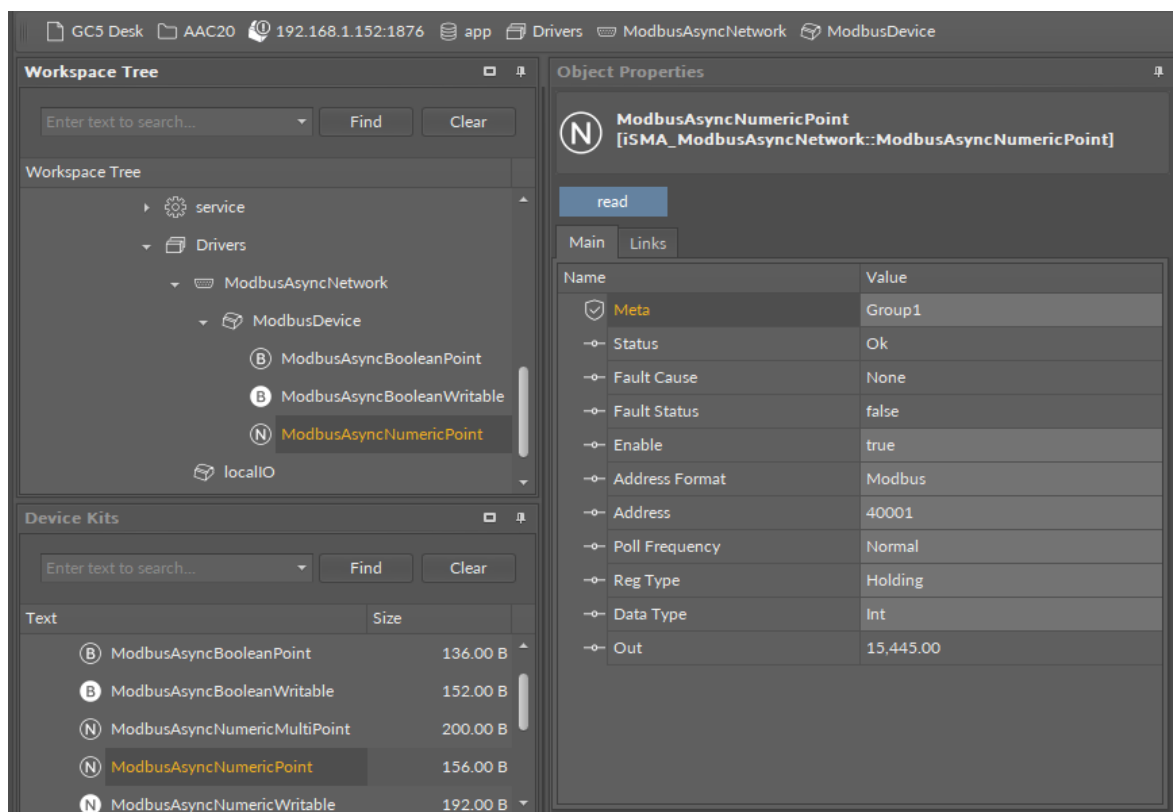


Figure 6. Modbus Async Numeric Point component

The ModbusAsyncNumericPoint component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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:** Enables/Disables the Point (true - Point enabled, false - Point disabled);
- **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 register 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 the read register.

4.4.4 ModbusAsyncNumericWritable Component

The ModbusAsyncNumericWritable Component 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:** Writes a value to the In slot and sends it to the device;
- **Write:** Sends a value from the In slot to the device;
- **Read:** Reads a value from the device and sends it to the Out slot.

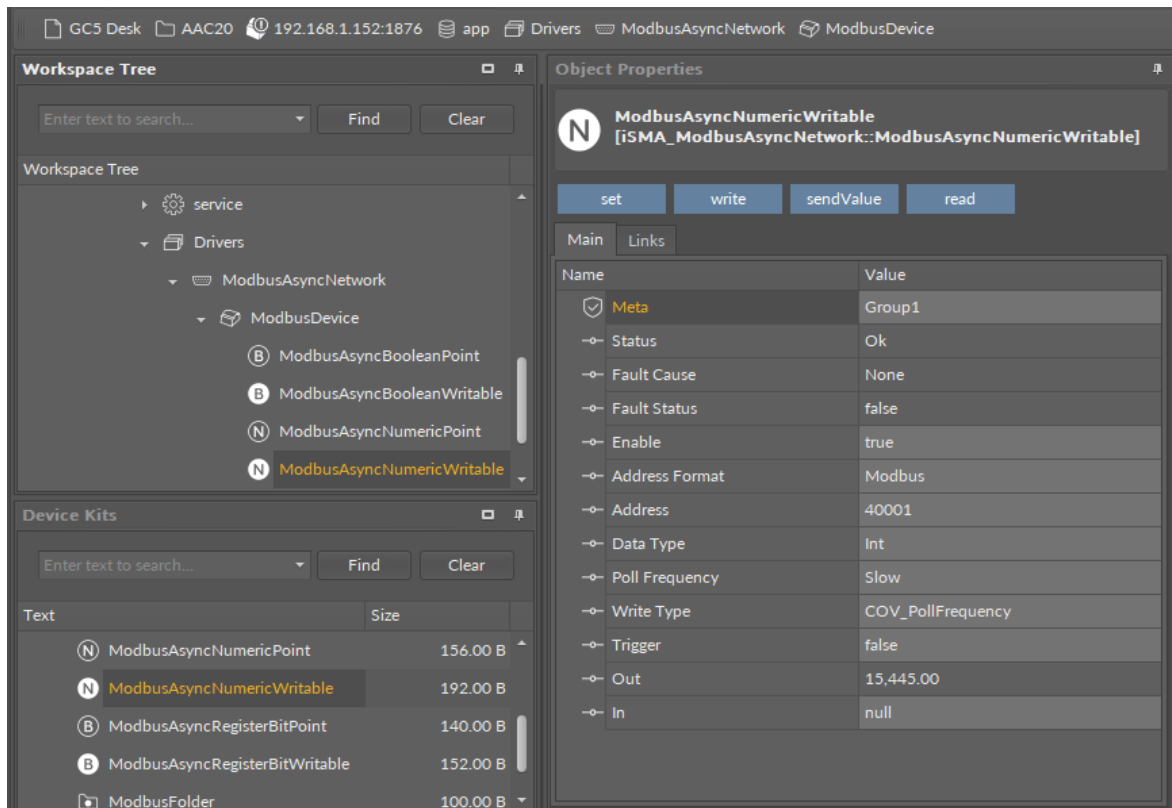


Figure 7. Modbus Async Numeric Writable component

The ModbusAsyncNumericWritable component has the following slots:

- **Status:** Network's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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 the 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 register 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 triggered by COV);
- **Trigger:** Forcefully send the value (on rising edge), regardless of the current poll mode,
- **Out:** Output slot, the current value of the device register,
- **In:** Input slot.

4.4.5 ModbusAsyncNumericMultiPoint

The ModbusAsyncNumericMultiPoint is a component which is responsible for reading up to eight 16-bits registers from the device in one message. The component uses 0x16 Modbus command. The component has a Read action available under the right-click, which forces the reading of the point.

The screenshot shows the configuration of the ModbusAsyncNumericMultiPoint component. The Object Properties panel displays the following configuration parameters:

Name	Value
Meta	Group1
Status	Ok
Fault Cause	None
Fault Status	false
Enable	true
Address Format	Modbus
Address	40001
Poll Frequency	Normal
Reg Type	Holding
Data Type	Int
Number Of Registers	8
Out1	15,445.00
Out2	1.00
Out3	7.00
Out4	109.00
Out5	0.00
Out6	6.00
Out7	0.00
Out8	109.00

Figure 8. Modbus Async Numeric Multi Point component

The ModbusAsyncNumericMultipoint component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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 the 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:** Enables/disables the Point (true - Point enabled, false - Point disabled);
- **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 registers read in one message;
- **Out:** Current value of the read register.

4.5 ModbusFolder Component

The ModbusFolder is a component which groups and organizes the Modbus points components. The ModbusFolder has the Description Slot where up to 32 characters may be inserted.

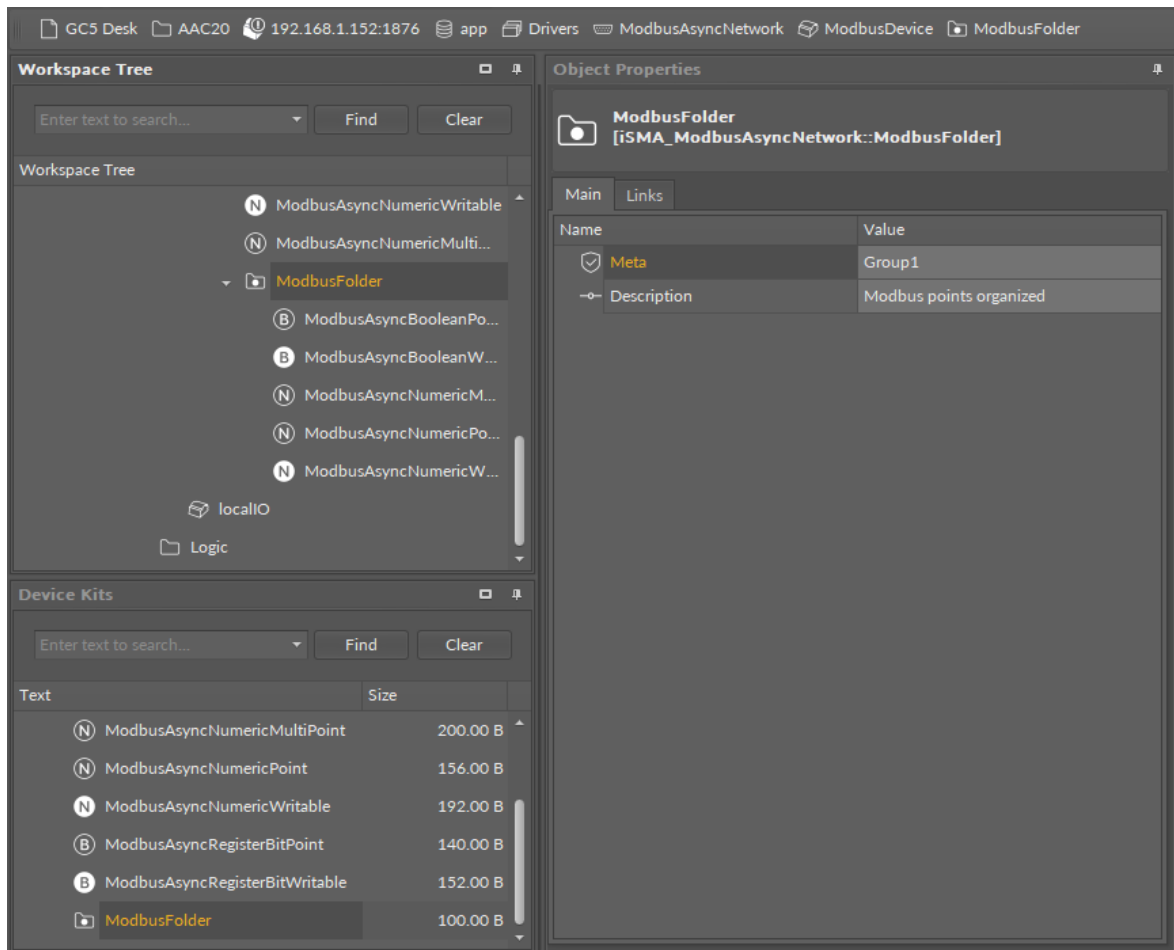


Figure 9. Modbus Folder component

5 Modbus TCP Network Kit

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

5.1 Modbus TCP License and Limitation

In a controller without the Modbus license there are available 15 data points. A license can be expanded up to 500 points by purchasing the Modbus license. The number of available points is shown in the ModbusTcpNetwork component in the 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 ModbusTCPNetwork Component

The ModbusTCPNetwork is the main component which is responsible for servicing an IP communication to slave devices. The component must be placed under the Drivers folder. The ModbusTCPNetwork sets parameters such as communication baud rate and data format, testing, etc., and keeps statistics.

The component has the following actions available under the right-click:

- **Reset Statistic:** Resets a network statistic value and starts counting from the beginning;
- **Enable/Disable:** Switches on/off the Modbus TCP network.

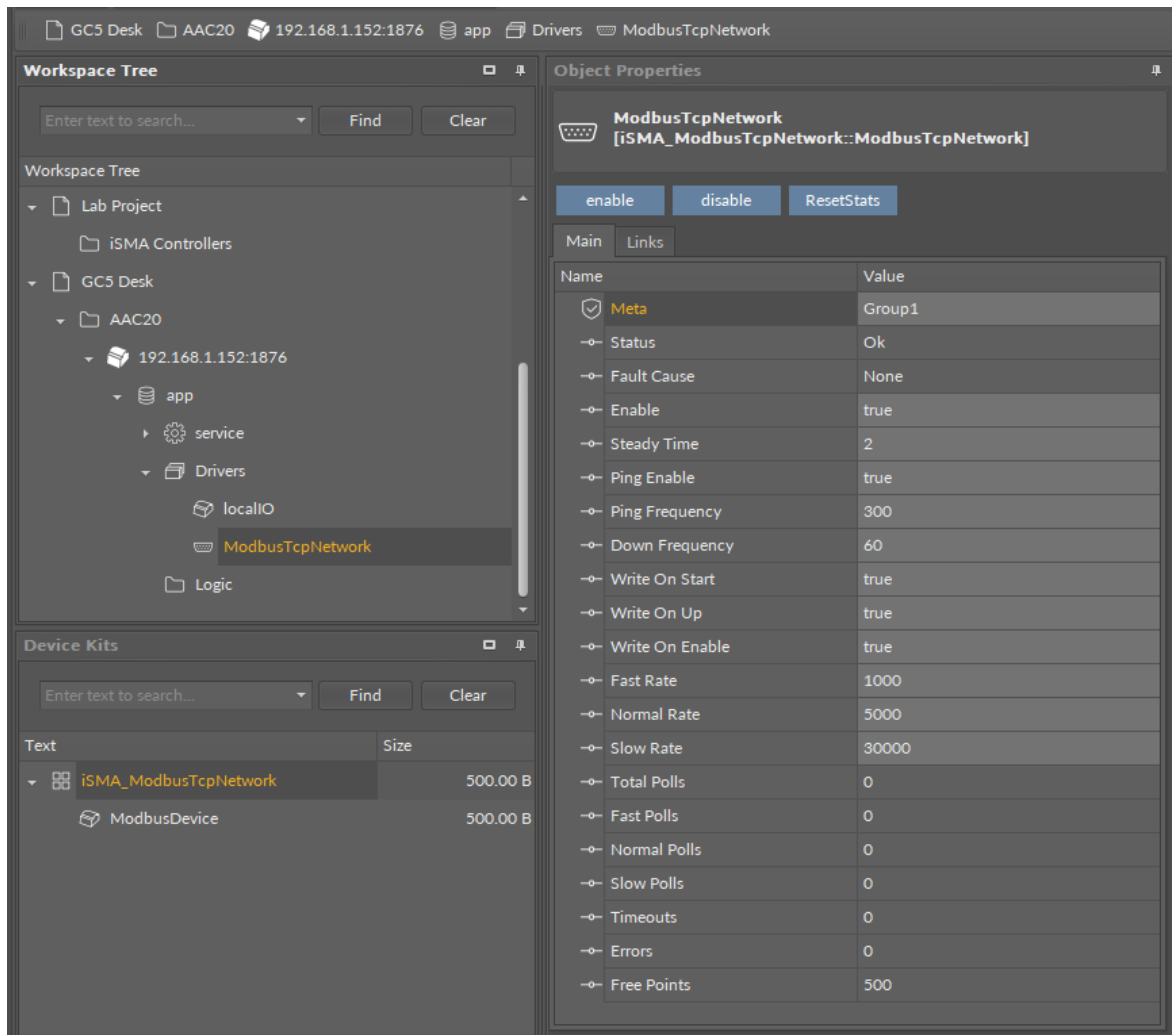


Figure 10. Modbus TCP Network component

The ModbusTCPNetwork component has the following slots:

- **Status:** Network's status, available states:
 - OK: Network is working properly,
 - Disabled: Network is disabled (Slot "Enable" is in false),
 - OK some device/point down: error in the device or points;
- **Fault Cause:** Fault cause description;
- **Enable:** Option to switch on or switch off Modbus network (true - Network enabled, false - Network disabled);
- **Steady Time:** Network's delay time to start-up after a power-up or reset;
- **Ping Enable:** Enables the device's connection testing function;
- **Ping Frequency:** Time between testing messages to check device connection;
- **Down Frequency:** Time between testing messages for devices or points which have got status down;

- **Write On Start:** Executes a write action in device „Writable” components in the Modbus Network after a reset or power-up;
- **Write On Up:** Executes a write action in device „Writable” components in the Modbus Network after restoring the connection with the Modbus device;
- **Write On Enable:** Executes a write action in device „Writable” components in the Modbus Network after enabling 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 of one message
- **Busy Time:** Percentage of Modbus Network usage;
- **Total Polls:** Total number of messages;
- **Fast Polls:** Number of messages sent in „Fast” mode;
- **Normal Polls:** Number of messages sent in „Normal” mode;
- **Slow Polls:** Number of messages sent in „Slow” mode;
- **Timeouts:** Number of lost messages, the difference between sent and received messages;
- **Errors:** Number of error messages (for example, with the wrong CRC);
- **Free points:** Number of available physical points in the Modbus Network.

5.3 ModbusTCPDevice Component

The ModbusDevice 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 a Ping action available under the right-click, which sends a test message to the device to check the 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. If any response is received from the device, including an exception response, this is considered a proof of communication, and the Modbus client device is no longer considered “down” if it had been previously marked “down”.

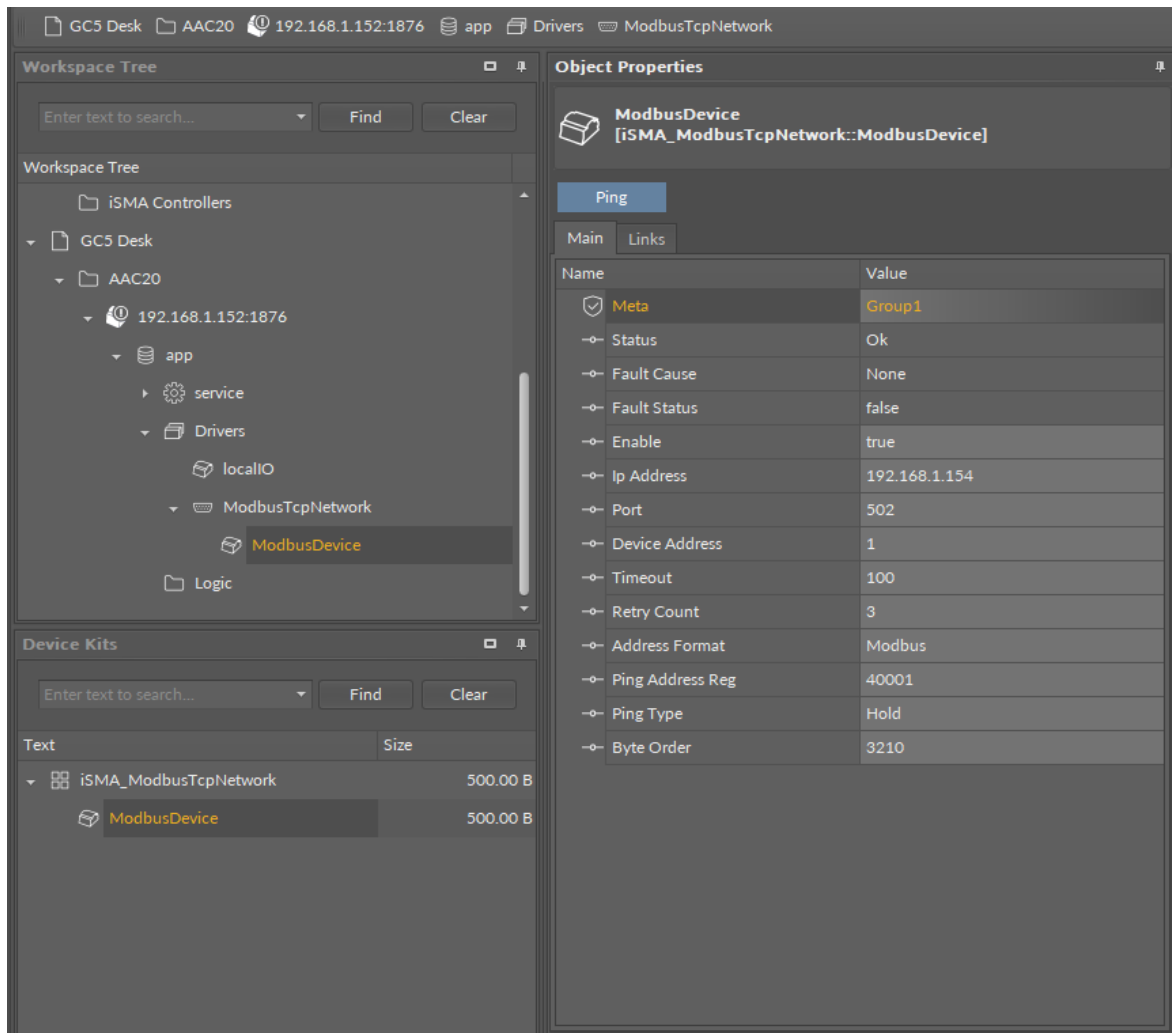


Figure 11. Modbus TCP Device component

The ModbusTCPDevice has the following slots:

- **Status:** Device's actual status (read-only), can take the following states:
 - OK: Device is working properly,
 - Disable: Device is disabled (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:** Enables/Disables the 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 the device request;

- **Inter Message Delay:** Time between messages sent to the device;
- **Retry Count:** Max. number of error messages (CRC error, lost messages);
- **Address Format:** Modbus address format (Modbus, Decimal);
- **Ping Address Reg:** Any register (Input/Holding) number for device connection test;
- **Ping Type:** Tested register type: Input/Holding;
- **Byte Order:** Byte reading order , for32-bit: 3210 (Big endian), 1032 (Little endian).

5.4 Modbus TCP Data Points

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

- **Boolean Point:** Reads Boolean values (Modbus command 0x02);
- **Boolean Writable:** Reads/writes Boolean values (Modbus command 0x05);
- **Numeric Point:** Reads numeric values (Modbus commands: Input - 0x04, Holding - 0x03);
- **Numeric Writable:** Reads/writes numeric values (Modbus commands: 16-bits Int, SInt - 0x06, 32-bits Long, SLong, Float: 0x16);
- **Numeric Multi Point:** Reads up to eight 16-bits registers (Modbus command 0x16).

5.4.1 ModbusTCPBooleanPoint Component

The ModbusTCPBooleanPoint is a component which is responsible for reading Boolean values from the device. The component has a Read action available under the right-click, which forces the reading of the point.

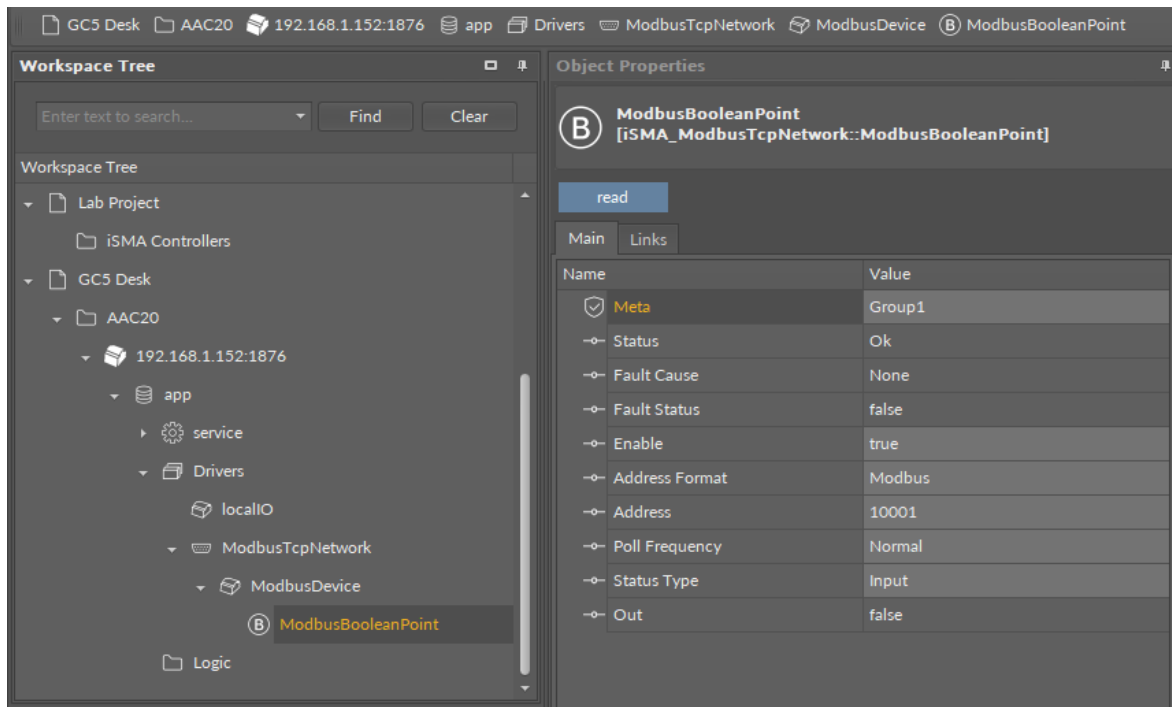


Figure 12. Modbus TCP Boolean Point component

The ModbusTCPBooleanPoint component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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:** Enables/disables the Point (true - Point enabled, false - Point disabled);
- **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 the read register;

5.4.2 ModbusTCPBooleanWritable Component

The ModbusTCPBooleanWritable is a component which is responsible for sending and reading Boolean values from the device.

The component has the following actions available under the right-click:

- **Set True/Set False:** Writes a value to the In slot and sends it to the device (not active when slot In have connected link),
- **Write:** Sends a value from the In slot to the device,
- **Read:** Reads a value from the device and send to the Out slot.

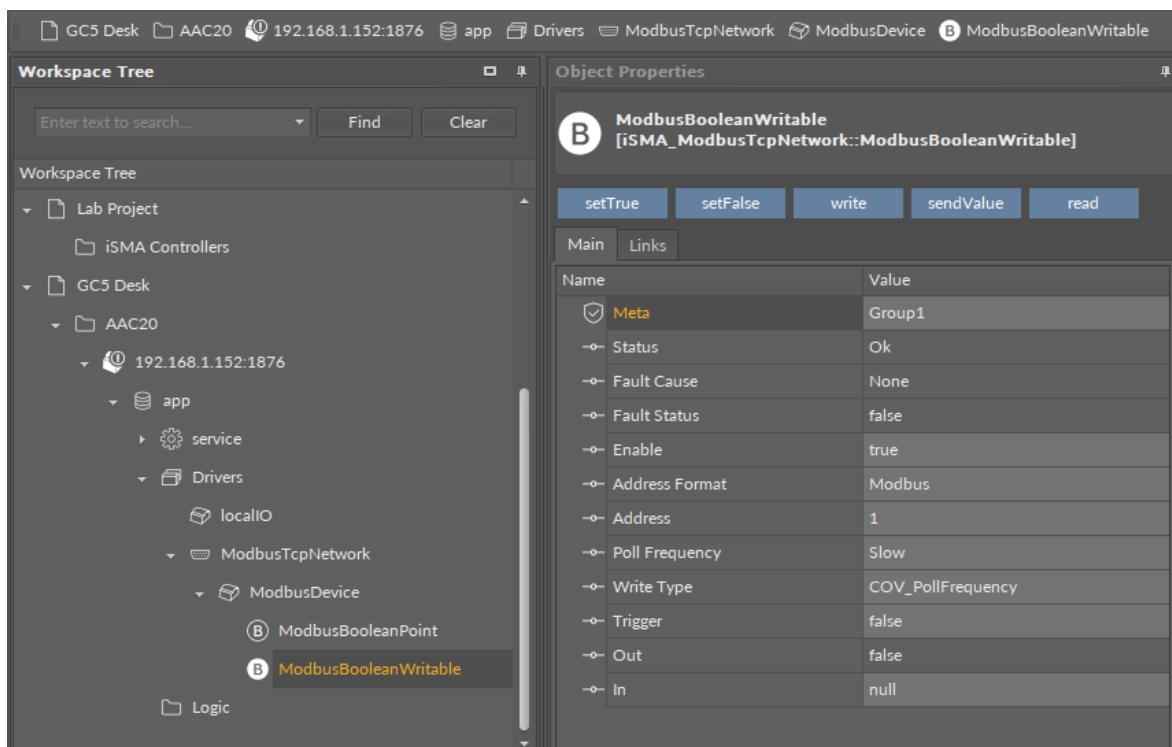


Figure 13. Modbus TCP Boolean Writable component

The ModbusBooleanWritable component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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:** Enables/disables the Point (true - Point enabled, false - Point disabled),
- **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 triggered by COV;
- **Trigger:** Forcefully send the value (on rising edge), regardless of the current poll mode,
- **Out:** Output slot, the current value of the read/write registry;
- **In:** Input slot.

5.4.3 ModbusTCPNumericPoint Component

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

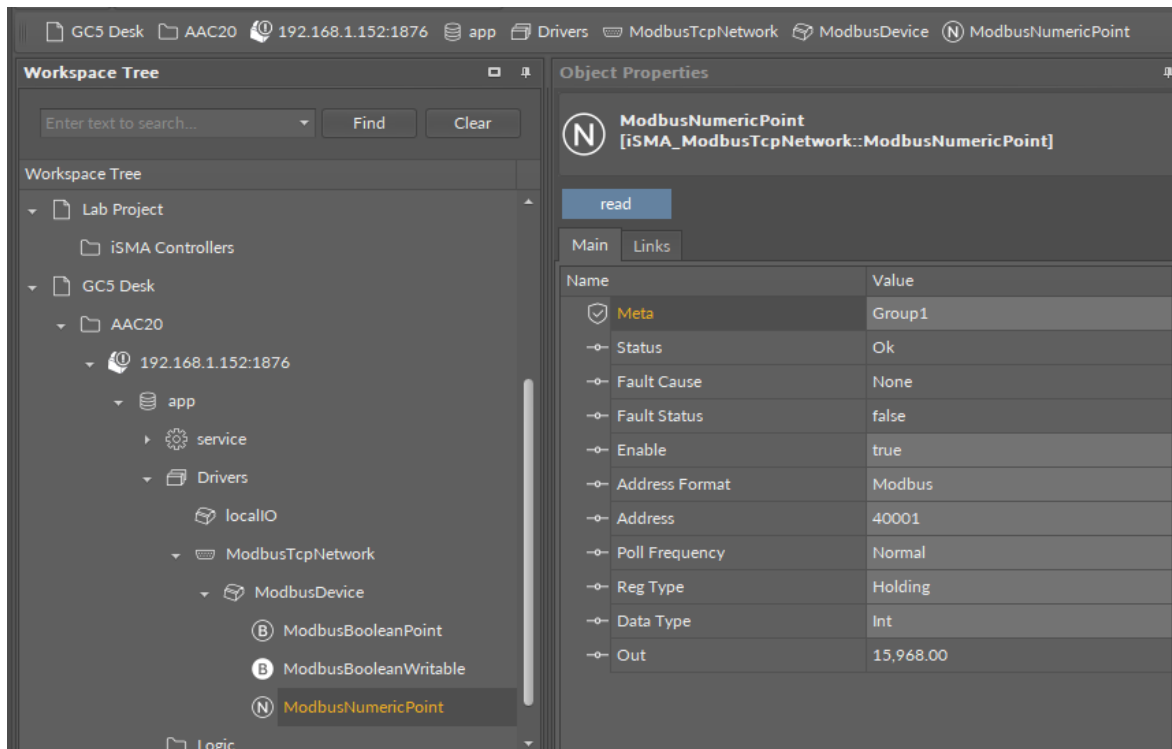


Figure 14. Modbus TCP Numeric Point component

The ModbusTCPNumericPoint component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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 the 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:** Enables/disables the Point (true - Point enabled, false - Point disabled);
- **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 register data type, available options: Int: 16-bits, Long: 32-bits, Float: 32-bits float-point, SInt: 16-bits with the sign, Slong: 32-bits with the sign;

- **Out:** Current value of the read register.

5.4.4 Modbus TCP Numeric Writable Component

The ModbusTCPNumericWritable 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-click button:

- **Set:** Writes a value to the In slot and sends it to the device;
- **Write:** Sends a value from the In slot to the device;
- **Read:** Reads a value from the device and sends it to the Out slot.

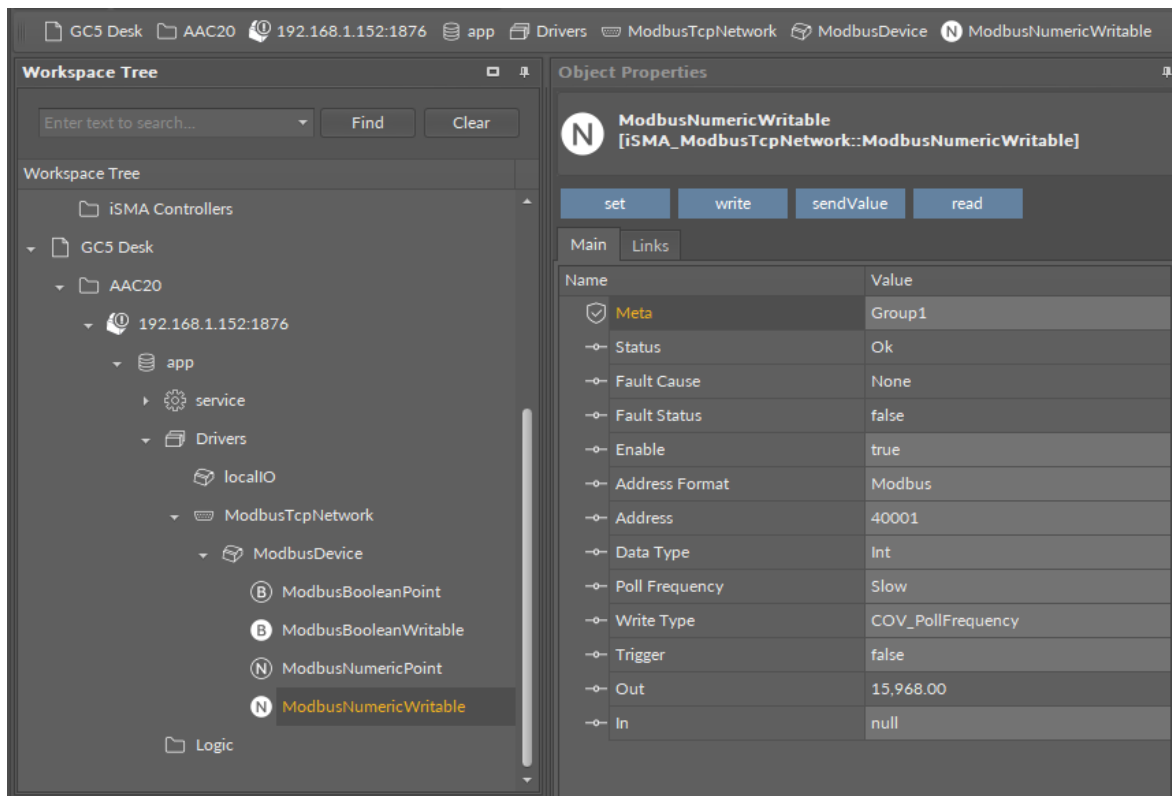


Figure 15. Modbus TCP Numeric Writable component

The ModbusNumericWritable component has the following slots:

- **Status:** Point's status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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 the 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 register 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 triggered by COV;
- **Trigger:** Forcefully send the value (on rising edge), regardless of the current poll mode;
- **Out:** Output slot, the current value of the device registry;
- **In:** Input slot.

5.4.5 ModbusTCPNumericMultiPoint Component

The ModbusTCPNumericMultiPoint is a component which is responsible for reading up to 8 registers from the device in one message. The component uses 0x16 Modbus command. The component has a Read action available under the right-click, which forces the reading of the point.

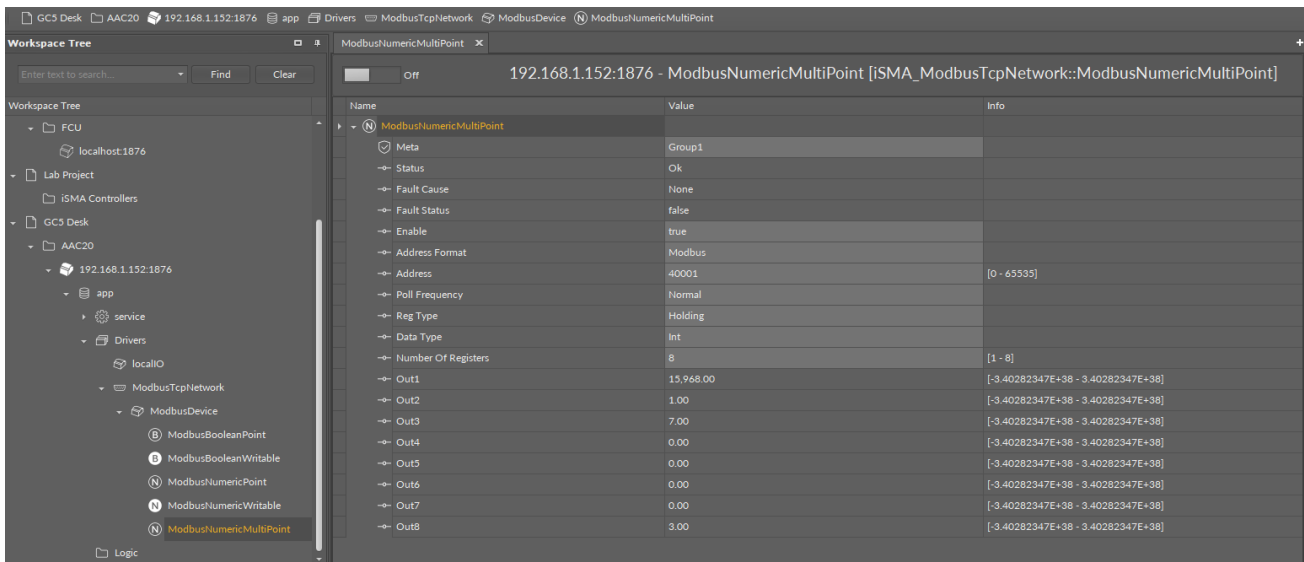


Figure 16. Modbus TCP Numeric Multipoint property sheet view

The ModbusTCPNumericMultiPoint component has the following slots:

- **Status:** Point's current status, available states:
 - OK: Point is working properly,
 - Disabled: Point is disabled (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:** Enables/disables the Point (true - Point enabled, false - Point disabled);
- **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, available options: Int (unsigned values), Sint (signed values);
- **Number Of Registers:** Number of the read registers in one message;
- **Out:** Current value of the read register.

5.5 ModbusFolder Component

The ModbusFolder is a component which groups and organizes the Modbus points components. The ModbusFolder has a Description Slot where up to 32 characters may be inserted.

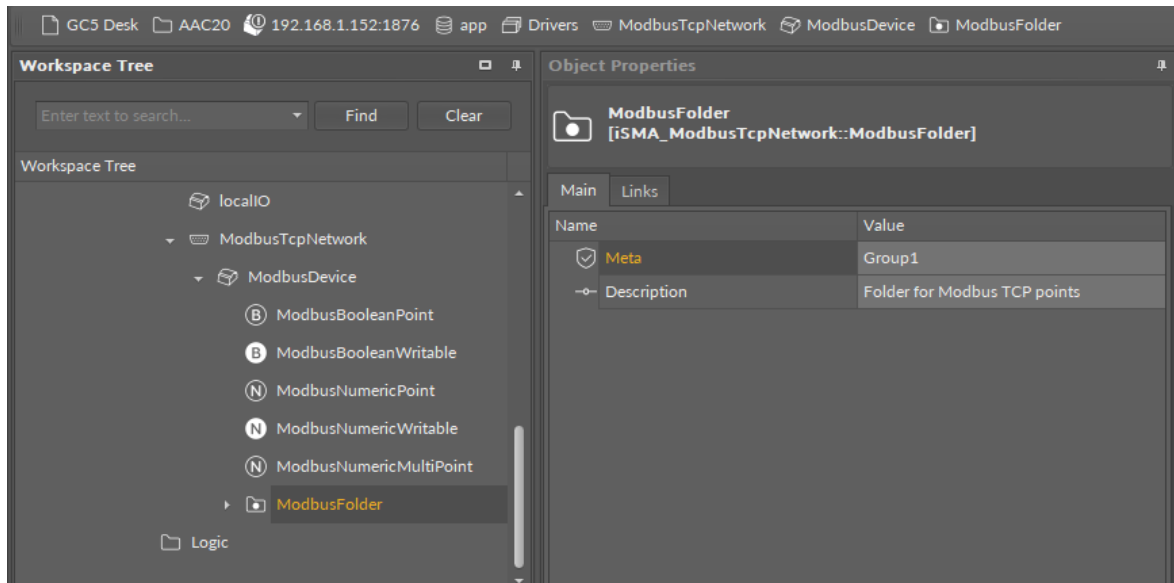


Figure 17. Modbus TCP Folder component

6 ModbusTCPSlaveNetwork Kit

The controller has a built-in Modbus TCP/IP Server (Modbus TCP Slave Device) application which is always active. The controller has a 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 the iSMA-B-AAC20 Modbus Table chapter. Addresses from 1000 to 2999 are free to use and can be adopted in the user application.

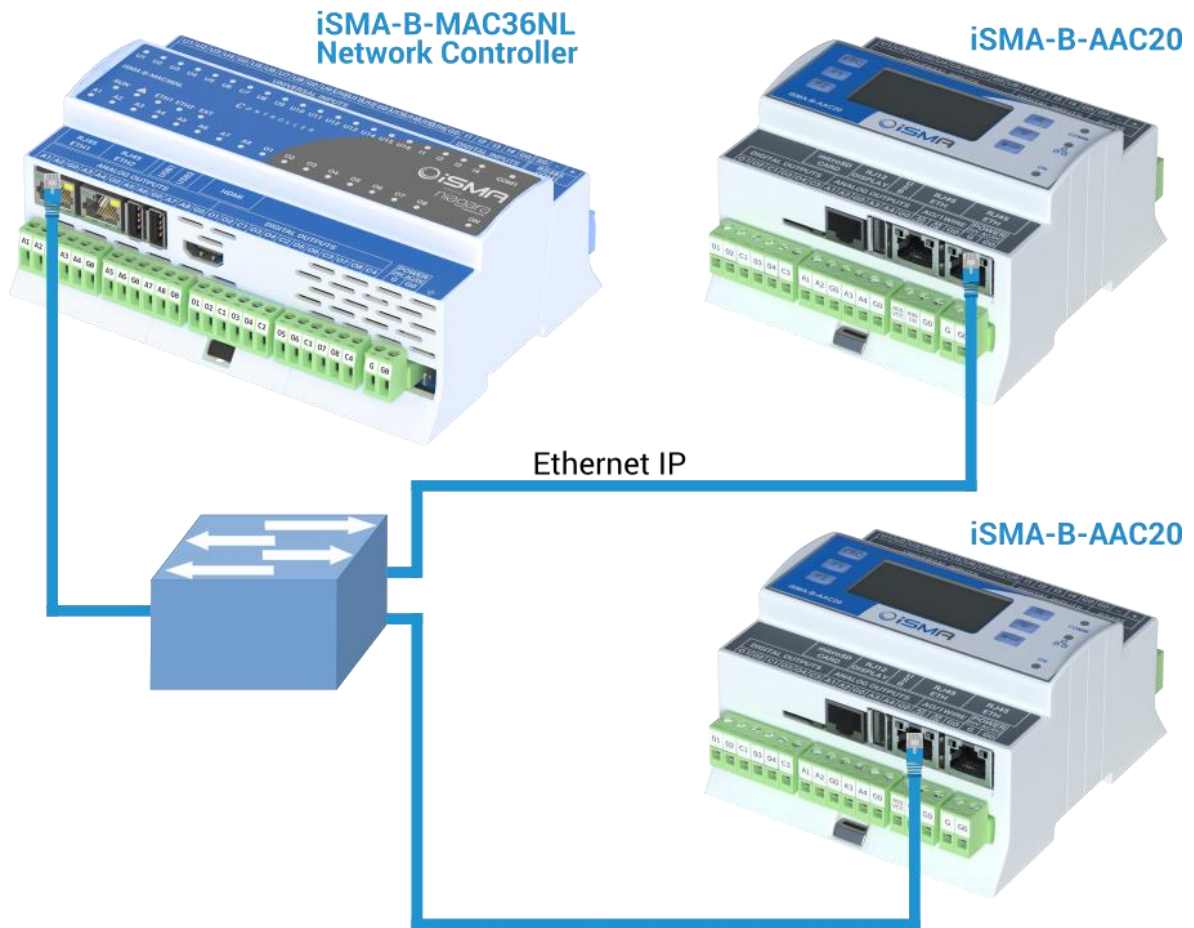


Figure 18. Modbus TCP topology

6.1 ModbusTCPSlaveNetwork

The ModbusTCPSlaveNetwork is always enabled, and it does not have to be configured in order to read the controller registers. The ModbusTCPSlaveNetwork is used only for changing parameters (parameters can be changed also from the controller's configuration web page) and to set up user registers.

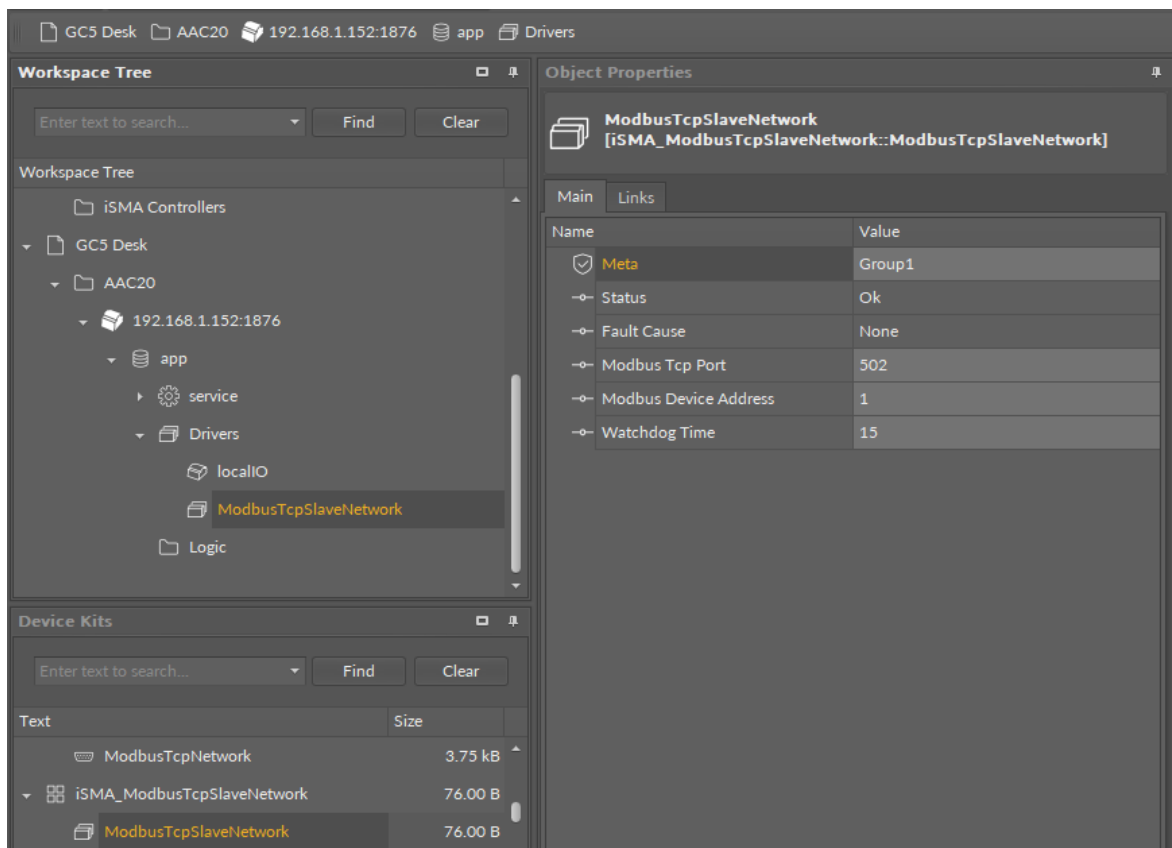


Figure 19. Modbus TCP Slave network component

The ModbusTCPSlaveNetwork has the following slots:

- **Status:** Network's 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 received valid messages, after which the controller will reset default values on outputs; value 0 disables this function.

6.2 Modbus TCP Slave Data Points

The Modbus TCP Slave Data Points are served by two components placed under the ModbusTCPSlaveNetwork component:

- **BooleanValue:** Reads/Writes Boolean values;
- **NumericValue:** Reads/Writes Numeric values.

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

WARNING! The BooleanValue and NumericValue both have the read and write function. To read-only use the Out slot only. Leave the In slot not connected with null (for Boolean) or nan (for Numeric) value.

WARNING! Using controller outputs in the Sedona application will disable writing function to the controller output registers. In this case, Sedona application has a higher priority.

6.2.1 BooleanValue Component

The BooleanValue component is responsible for reading and writing Boolean values in the controller's Modbus table. Values can be read only for Modbus Master (Bit Type: Discrete Input) or read and write for Modbus Master (Bit Type: Coil).

Only one Modbus table is used for writing the Boolean and Numeric values. Before addressing the component make sure that the register is not in use by any another component.

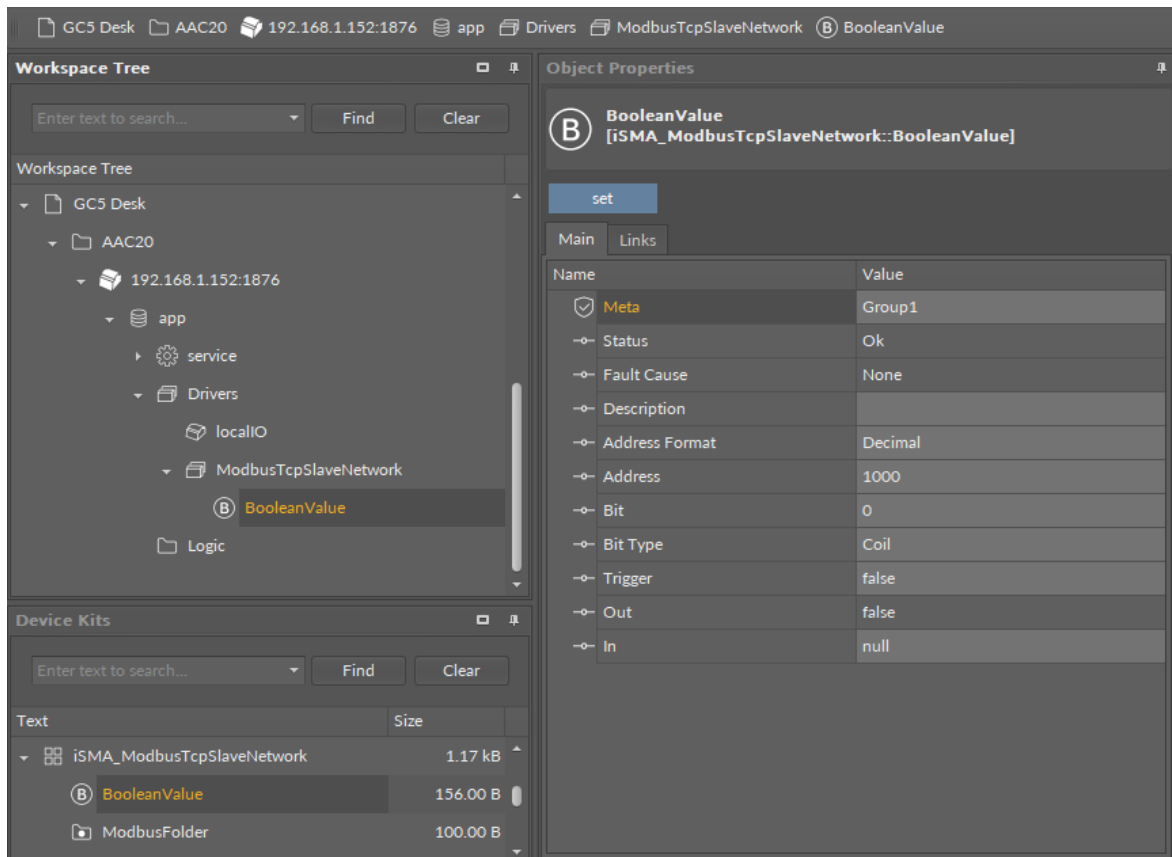


Figure 20. Modbus TCP Slave Boolean value component

The Modbus TCP Slave BooleanValue component has the following slots:

- **Status:** Point's current status;
- **Fault Cause:** Fault cause description;
- **Description:** Point description label up to 32 characters;
- **Address Format:** Modbus addressing format: Modbus/Decimal,
- **Address:** Register address (from 0 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:** Forcefully send the Input value to controller Modbus table (on rising edge);
- **Out:** Output slot, the current value of the device register;
- **In:** Input slot.

6.2.2 NumericValue Component

The NumericValue component is responsible for reading and writing numeric values to the controller's Modbus table. Values can be read only for the Modbus Master (Register Type: Input Register) or read and write for the Modbus Master (Register Type: Holding Register).

Only one Modbus table is used for writing the Boolean and Numeric values. Before addressing the component make sure that the register is not in use by any another component.

WARNING! Data Types: Long, SLong, and Float use 32-bits format and use two registers. Next free register in the table is the Register address + 2.

For example: Float value register address is 1010, the next register must be addressed 1012.

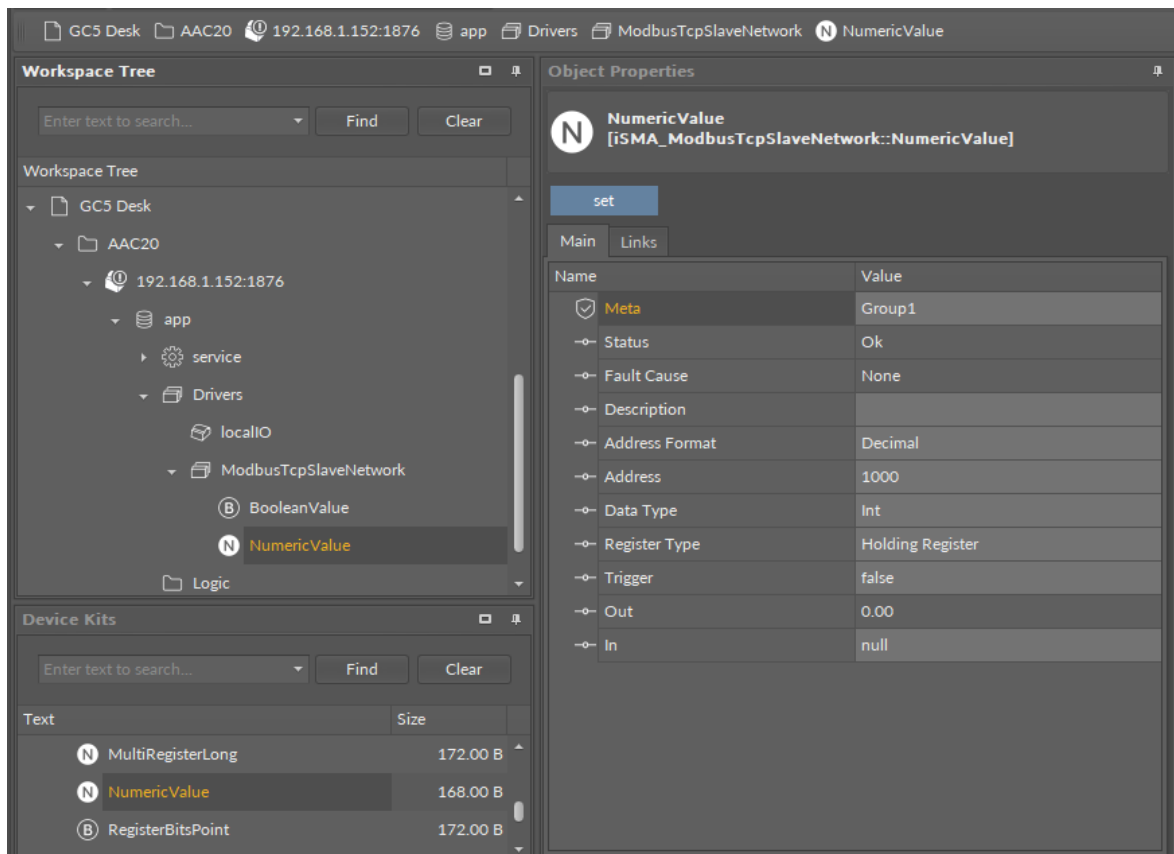


Figure 21. Modbus TCP Slave Numeric value component

The Modbus TCP Slave NumericValue component has the following slots:

- **Status:** Point's status;
- **Fault Cause:** Fault cause description;
- **Description:** Point description label up to 32 characters;
- **Address Format:** Modbus addressing format: Modbus/Decimal;
- **Address:** Register address (from 0 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:** Forcefully send the Input value to controller Modbus table (on rising edge);
- **Out:** Output slot, the current value of the device register;
- **In:** Input slot.

6.3 ModbusFolder Component

The ModbusFolder is a component which groups and organizes the Modbus points components. The ModbusFolder has a Description Slot where up to 32 characters may be inserted.

7 Modbus RJ12 Network Kit

The iSMA-B-AAC20 controller has one RJ12 port which can be used as a Modbus RTU/ASCII Master.

7.1 Modbus RJ12 License and Limitation

In a standard license there are available 500 data points, and this number cannot be expanded. The number of available points is shown in the ModbusRJ12Network component in the Free Points slot.

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

7.2 ModbusRJ12Network Component

The ModbusRJ12Network is the main component which is responsible for servicing an RJ12 physical port. The component must be placed under the Drivers folder. The ModbusRJ12Network sets parameters such as communication baud rate and data format, testing, etc., and keeps statistics.

The component has the following actions available under the right-click or in the Object Properties window:

- **Reset Stats:** Resets the network's statistics and starts counting from the beginning,
- **Enable/Disable:** Switches on/off the ModbusRJ12Network.

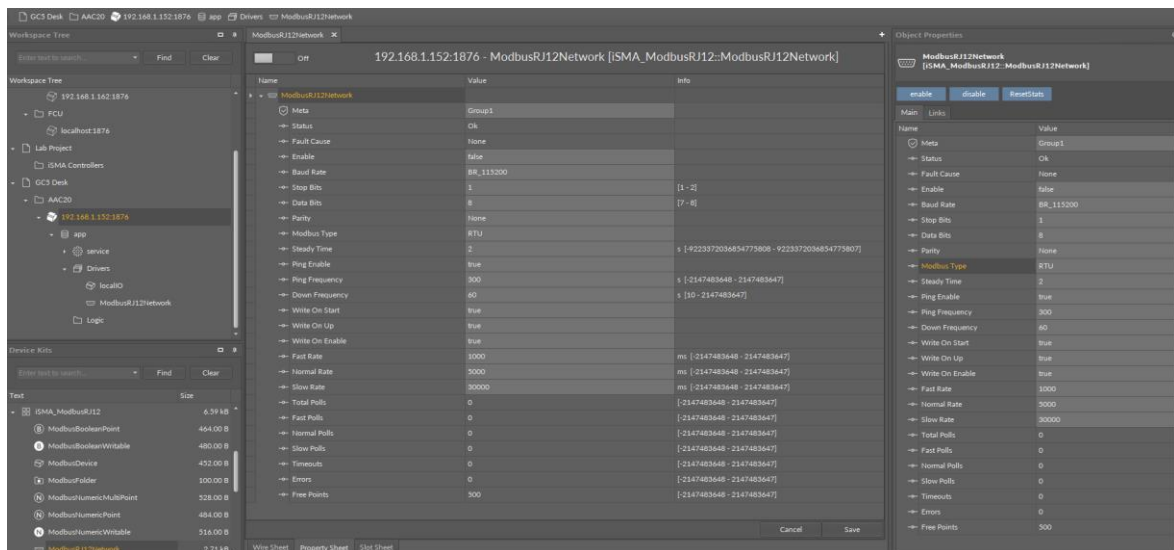


Figure 22. Modbus RJ12Network component

The ModbusRJ12Network component has the following slots:

- **Status:** Network's current status, available states:
 - OK: Network is working properly,
 - Disabled: Network is disabled (Slot "Enable" is in false),
 - OK some device/point down: error in device or points;
- **Fault Cause:** Fault cause description;
- **Enable:** Switches on or switches off the ModbusRJ12Network (true - Network enabled, false - Network disabled);
- **Steady Time:** Network start-up delay time after a 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:** Enables the device's connection testing function;
- **Ping Frequency:** Time between testing messages to check the device connection;
- **Down Frequency:** Time between testing messages for devices or points which have got status down;
- **Write On Start:** Executes a write action in device „Writable“ components in the ModbusRJ12Network after a reset or power-up;
- **Write On Up:** Executes a write action in device „Writable“ components in the ModbusRJ12Network after restoring the connection with the Modbus device,
- **Write On Enable:** Executes a write action in device „Writable“ components in the ModbusRJ12Network after enabling 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;
- **Total Polls:** Total number of messages;
- **Fast Polls:** Number of messages sent in „Fast“ mode;
- **Normal Polls:** Number of messages sent in „Normal“ mode;
- **Slow Polls:** Number of messages sent in „Slow“ mode;
- **Timeouts:** Number of lost messages, a difference between sent and received messages;
- **Errors:** Number of error messages (for example, with the wrong CRC);
- **Free points:** Number of available physical points in the ModbusRJ12Network.

7.3 ModbusRJ12Device Component

The ModbusRJ12Device is a component which is responsible for servicing a physical device connected to the ModbusRJ12Network. The device is a Modbus Master to all other Modbus devices on the attached RJ12 port. Each device is represented by the ModbusDevice, and has a unique Modbus address (from 1 to 247) as well as the other Modbus config data and starting addresses for Modbus data items (coils, inputs, input registers, holding registers). The component has a Ping action available under the right-click, which sends a test message to the device to check the 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 Enable). When enabled the network monitor periodically pings (queries) this address. If any response is received from the device, including an exception response, this is considered a proof of communication, and the Modbus client device is no longer considered “down” if it had been previously marked “down”.

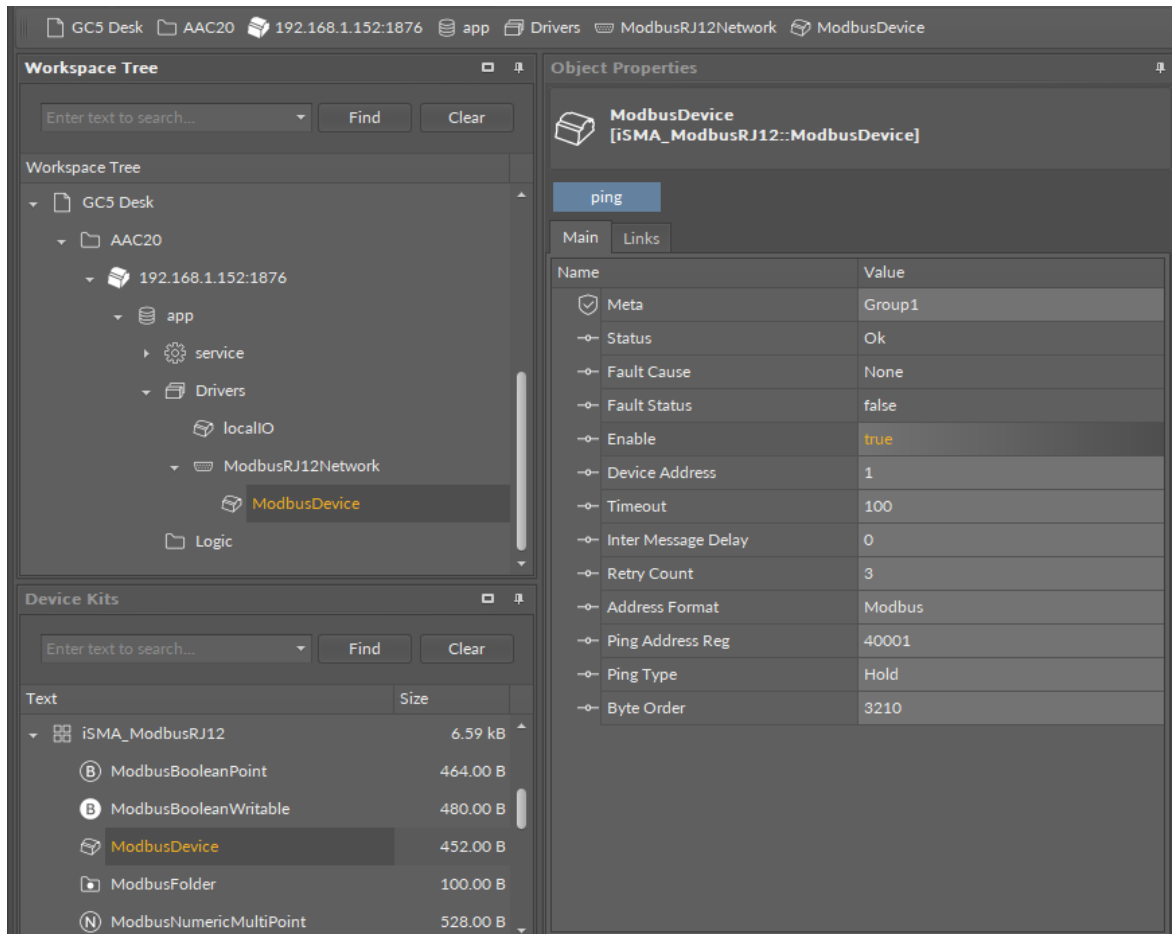


Figure 23. Modbus RJ12 Device component

The ModbusRJ12Device has the following slots:

- **Status:** Device’s current status (read-only), can take the following states:
 - OK: Device is working properly,
 - Disable: Device is disabled (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:** Enables/Disables the device;

- **Device Address:** Modbus device physical address (0 - network broadcast address, 1-248 addressing range);
- **Timeout:** Max. device response time from the device request;
- **Inter Message Delay:** Time between messages sent to the device;
- **Retry Count:** Max. number of error messages (CRC error, lost messages);
- **Address Format:** Modbus address format (Modbus, Decimal);
- **Ping Address Reg:** Any register (Input/Holding) number for device connection test;
- **Ping Type:** Tested register type: Input/Holding;
- **Byte Order:** Byte reading order, for32-bit: 3210 (Big endian), 1032 (Little endian).

7.4 Modbus Data Points

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

- **Boolean Point:** Reads Boolean values (Modbus command 0x02);
- **Boolean Writable:** Reads/writes Boolean values (Modbus command 0x05);
- **Numeric Point:** Reads numeric values (Modbus commands: Input - 0x04, Holding - 0x03);
- **Numeric Writable:** Reads/writes numeric value (Modbus commands: 16-bits Int, SInt - 0x06, 32-bits Long, SLong, Float: 0x16);
- **Numeric Multi Point:** Reads up to eight 16-bits registers (Modbus command 0x16).

The ModbusRJ12Network uses the same data points as the ModbusAsyncNetwork. For detailed information please refer to point 4.4 of this manual.

7.5 Modbus RJ12 Wiring

The RJ12 port connector is located between the SD card and USB slots. The connector provides Modbus bus wires, ground potential G0, and power supply directly connected to G terminal from the power supply connector (the external devices can be powered through the RJ12 connector). A wiring diagram is shown in the figure below.

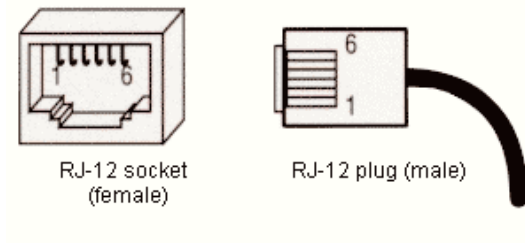


Figure 24. RJ12 connector pins numbers

The RJ12 pins description:

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

Connection of the devices is shown in the figure below.

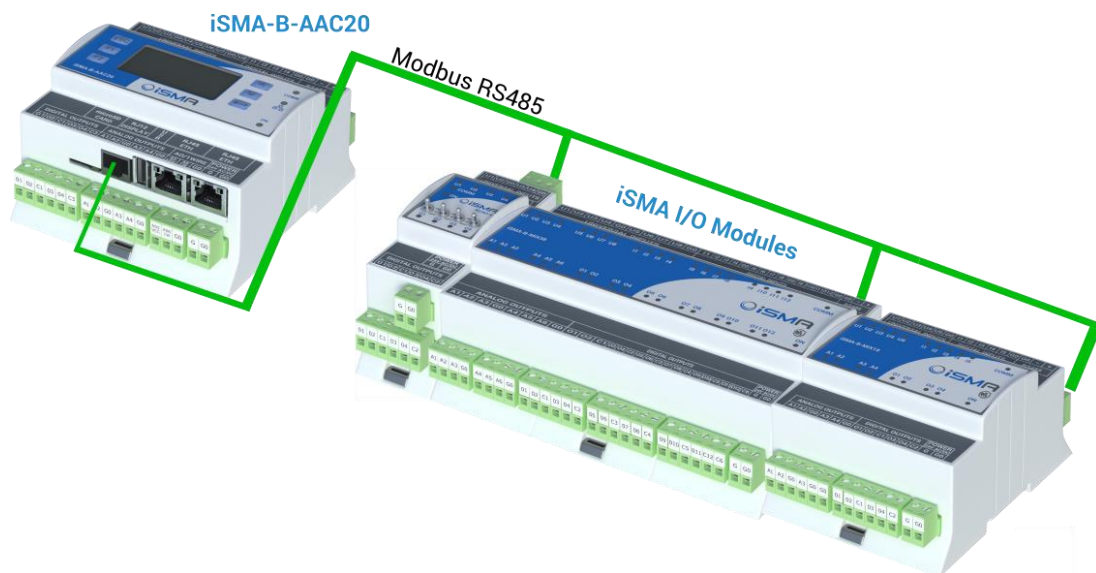


Figure 25. RJ12 to RS485 Modbus connection

8 Gateway Mode

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

The RS485 communication parameters can be set up by:

- Controller configuration Web Page (RS485 Configuration tab);
- ModbusAsyncNetwork component (remember to set false in the Enable slot);
- Controller Modbus register table (see the iSMA-B-AAC20 Modbus table chapter).

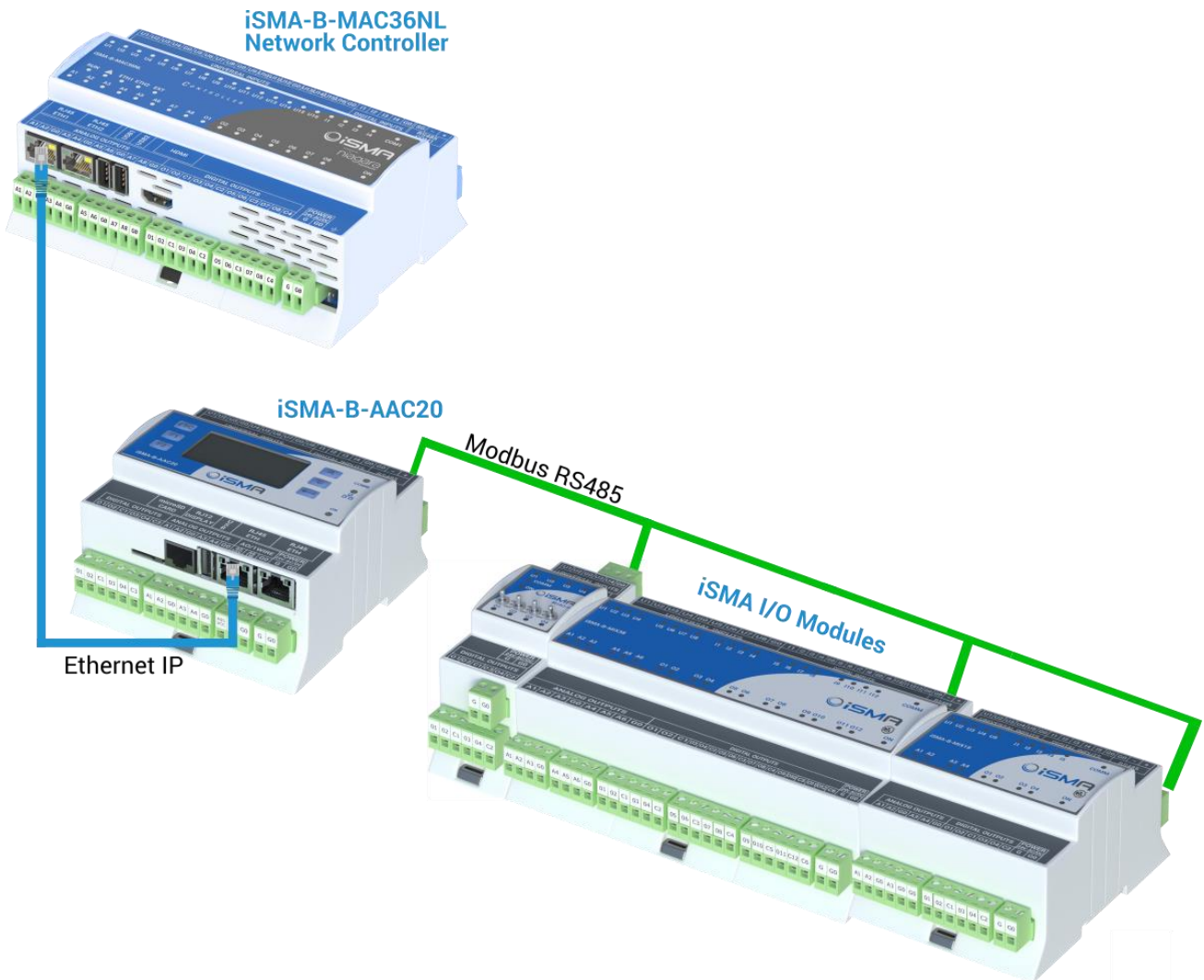


Figure 26. Modbus TCP gateway topology

9 iSMA Module

The iSMA Modules is an extension of the ModbusAsyncNetwork designed to easily serve iSMA devices series like MIX modules, MINI modules and Wireless Module using Modbus ASCII/RTU protocol. The iSMA Modules kit contains prepared components for serving physical inputs, outputs, and configuration parameters. The iSMA Modules kit consists of 4 types of components:

- ModbusNetwork;
- iSMADevice;
- iSMAIOPoints;
- iSMADeviceConfig;
- iSMAPointsFolder.

WARNING! The iSMA Device must be placed under the ModbusNetwork component from the ModbusAsyncNetwork kit.

9.1 iSMADevice Component

The iSMADevice is a component designed to cooperate with iSMA devices hardware. This component has built-in parameters to work with all iSMA devices in the ModbusAsyncNetwork.

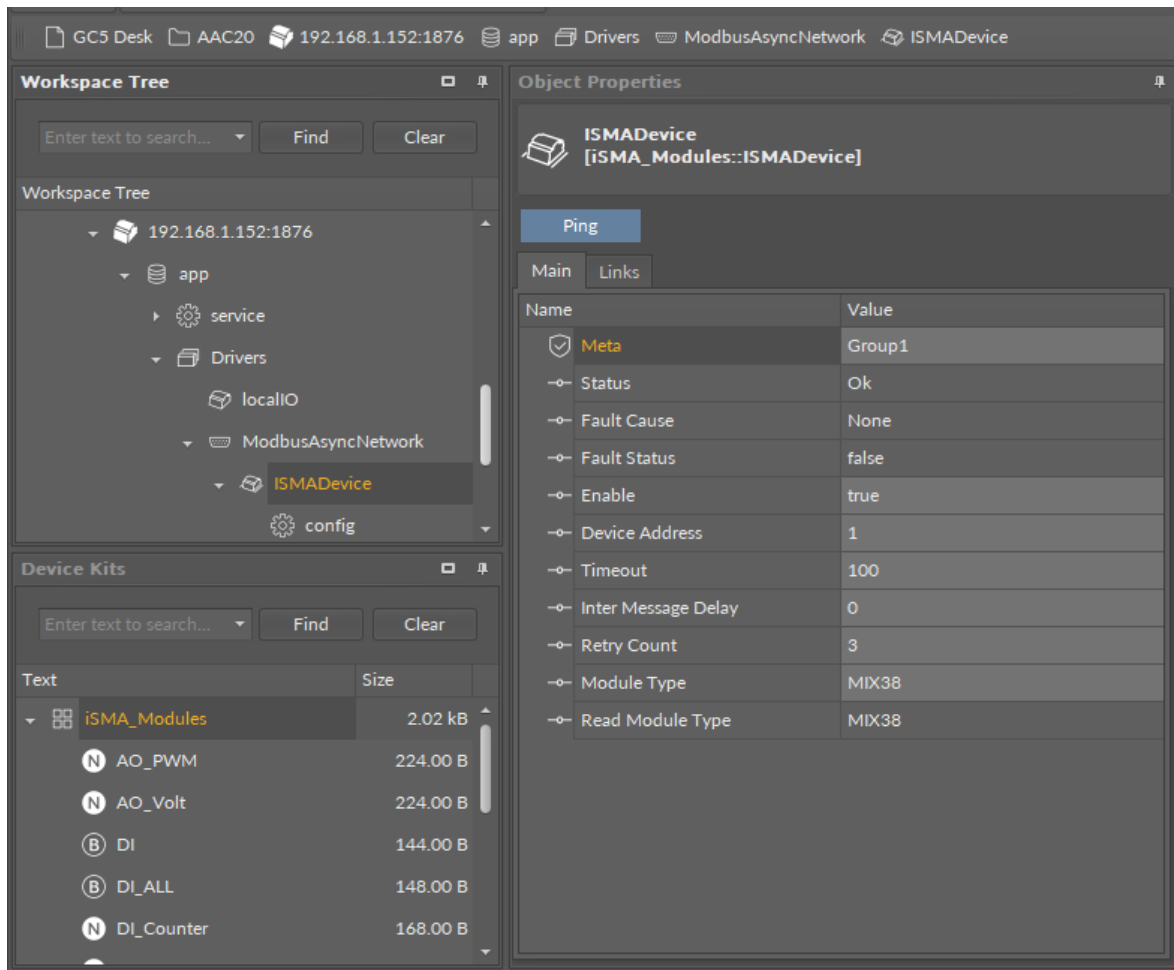


Figure 27. iSMA Module Device component

9.2 iSMADeviceConfig Component

The iSMADeviceConfig is a special component dedicated to set up the iSMA series device parameters. Adding and removing of the iSMADeviceConfig component is done by the Module Type slot in the iSMADevice component. To add a configuration component a proper module type is selected from a drop-down list. When the connection is established, the module type is displayed in the Read Module Type slot.

WARNING! This component has no auto-refresh option. To read or to write the device configuration the component action must be used. It is recommended to read module configuration before changing parameters.

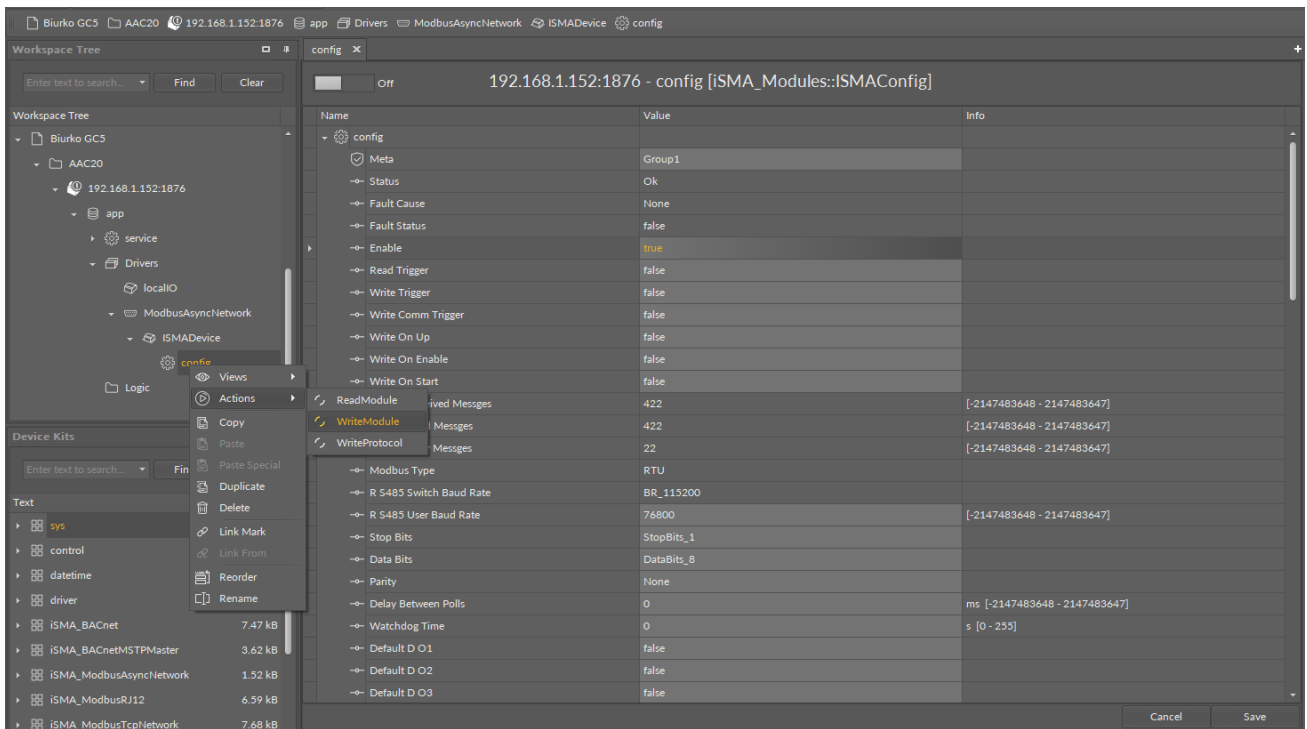


Figure 28. iSMA Module config component

9.3 iSMA IO components

iSMA Module was created to simplify a commissioning process for iSMA MIX, MINI, and Wireless series devices. iSMA Module kit contains components to serve all inputs/outputs.

9.3.1 DI Components

In the iSMA Module kit there are available two types of components to read Digital Inputs:

- **DI:** Reads individual Digital Inputs (input number is selected in the component property sheet);
- **DI_ALL:** Reads all Digital Inputs using one register.

Note: DI_ALL component has 12 input slots (DI component - up to 12 inputs) which correspond to the largest module iSMA-B-MIX38. Using a module with a smaller number of inputs makes the surplus inputs inactive and always in a false state.

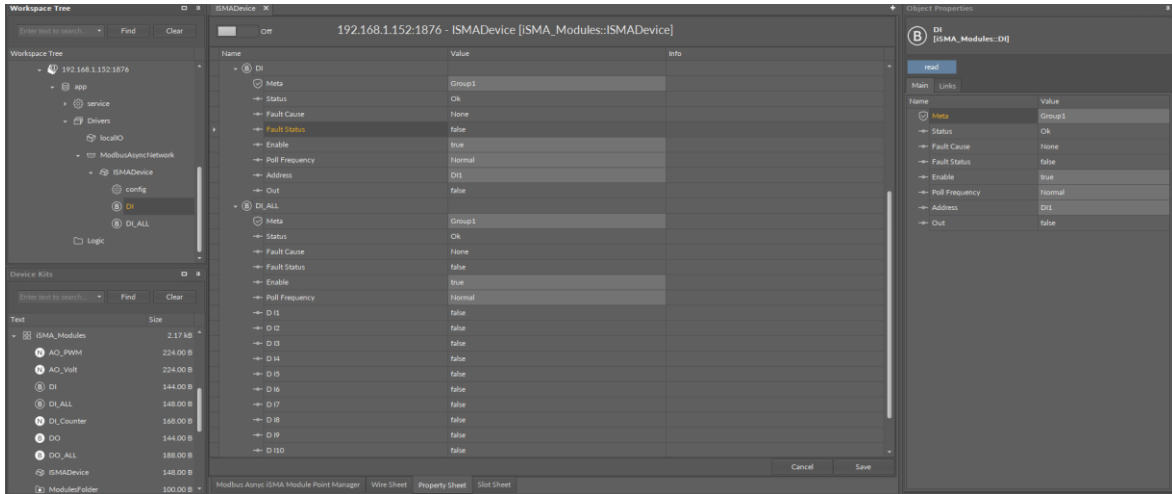


Figure 29. iSMA Module Digital Input components

9.3.2 DO Components

In the iSMA Module kit there are available two types of components to read/write Digital Outputs:

- **DO:** Reads/writes individual Digital Outputs (output number is selected in component property sheet);
- **DO_ALL:** Reads all Digital Outputs using one register.

Note: DO_ALL component has 12 input slots (DO component - up to 12 outputs) which correspond to the largest module iSMA-B-MIX38. Using module with a smaller output number makes the surplus outputs inactive.

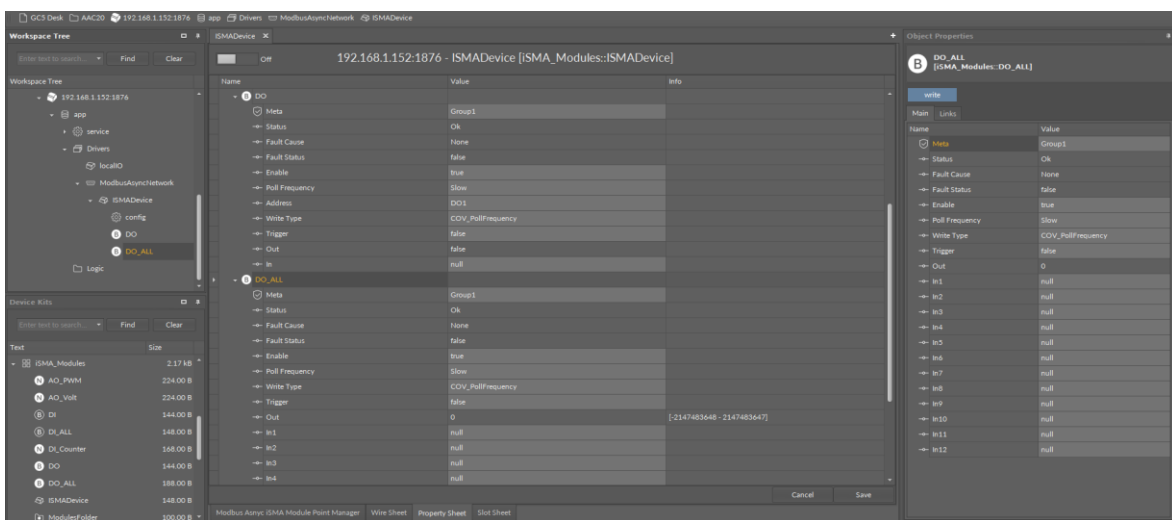


Figure 30. iSMA Module Digital Output components

9.3.3 UI components

In the iSMA Module kit there are available five types of components to read Universal Inputs:

- **UI_Temp:** Reads a temperature value from the NTC sensor connected to the Input;
- **UI_Res:** Reads a resistance value between the Universal Input and G0;
- **UI_Volt:** Reads a voltage value between the Universal Input and G0;
- **UI_DI:** Reads a Boolean value (dry contact) from a single Universal Input;
- **UI_DI_ALL:** Reads a Boolean value (dry contact) from all Universal Inputs in one register.

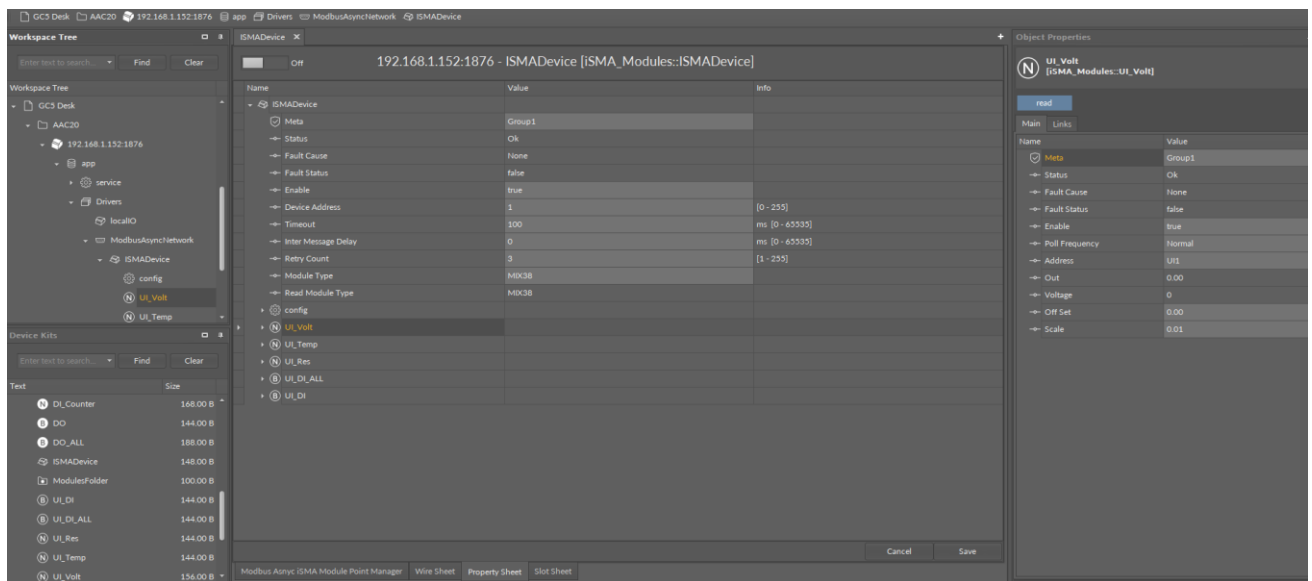


Figure 31. iSMA Module Universal Input components

The Universal Inputs configuration (sensor type, voltage measurement, filter time, and resolution) is done in the iSMADeviceConfig component.

The input number is selected in the 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.

9.3.4 AO components

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

- **AO_Volt:** to set up a voltage signal (0 – 10000mV) on the Analog Output;
- **AO_PWM:** to set up a PWM signal (0 – 100%).

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

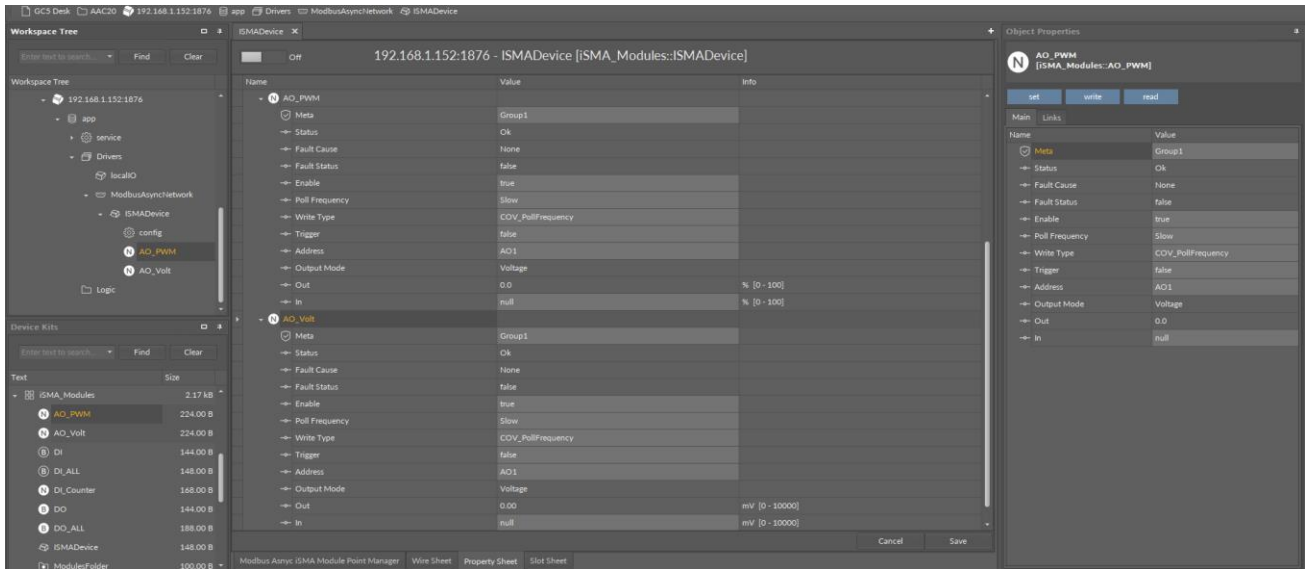


Figure 32. Analog Outputs property sheet view

The Input number is selected in the component property sheet, sensor type - in the module config component. Notice that AO components have 6 inputs, which correspond to the largest module iSMA-B-MIX38. Using a module with a smaller output number makes the surplus outputs inactive.

9.3.5 iSMAFolder Component

The iSMAFolder is a component which groups and organizes the iSMA Module IO point components. Because of Sedona components, names are limited to 7 characters, ModbusFolder has a Description Slot which can use up to 32 characters.

10 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 address												
40009	8	0x08	RS485 BAUD RATE	Read & Write Memory	Transmission baud rate is defined by the user calculated using the formula: $\text{Baudrate} = (\text{registervalue}) \cdot 10$ The default value is 11520 (115200 bps)												
40010	9	0x09	RS485 STOP BITS	Read & Write Memory	Supported values are 1 and 2 The default value is 1												
40011	10	0x0A	RS485 DATA BITS	Read & Write Memory	Supported values are 7 and 8 The default value is 7												
40012	11	0x0B	RS485 PARITY BIT	Read & Write Memory	The default value is 0 (no parity) Allowed values: <table border="1" data-bbox="1002 1391 1442 1899"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table>	Value	Description	0 (default)	none	1	Odd	2	Even	3	Always 1	4	Always 0
Value	Description																
0 (default)	none																
1	Odd																
2	Even																
3	Always 1																
4	Always 0																
40014	13	0x0D	RS485 MODBUS PROTOCOL TYPE	Read & Write	Protocol Type (40014) 0 – RTU, 1 – ASCII												

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description	
				Memory	The default value is 0 - RTU.	
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	32-bit counters for each Digital Input counting pulses.	
40024	23	0x17	COUNTER 1 MSB	Memory		
40025	24	0x18	COUNTER 2 LSB	Read & Write		
40026	25	0x19	COUNTER 2 MSB	Memory		
40027	26	0x1A	COUNTER 3 LSB	Read & Write		
40028	27	0x1B	COUNTER 3 MSB	Memory		
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.
30072	71	0x47	UNIVERSAL INPUT TEMPERATURE 1	Read-only		

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description
30073	72	0x48	UNIVERSAL INPUT VOLTAGE 2	Read-only	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 For a result, divide the registry value by 10: $T = \frac{\text{registervalue}}{10}$ Selection of the type sensor is done using UNIVERSAL INPUT CONFIGURATION register from 40151 to 40158 for each input separately
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	
30081	80	0x50	UNIVERSAL INPUT VOLTAGE 6	Read-only	
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	

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description
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	<p>Resistance measurement result expressed in Ω</p> <p>Value range from 0 Ω to 1 000 000 Ω</p> <p>Note: In PT1000 or NI1000 input working type the reading accuracy increase and the register value is multiply by 10</p>
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	

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description
30113	112	0x70	RESISTIVE INPUT 6 LSB	Read-only	
30114	113	0x71	RESISTIVE INPUT 6 MSB	Read-only	
30115	114	0x72	RESISTIVE INPUT 7 LSB	Read-only	
30116	115	0x73	RESISTIVE INPUT 7 MSB	Read-only	
30117	116	0x74	RESISTIVE INPUT 8 LSB	Read-only	
30118	117	0x75	RESISTIVE INPUT 8 MSB	Read-only	
40121	120	0x78	VALUE OF ANALOG OUTPUT 1	Read & Write	The voltage at the Analog Outputs are given in the mV range from 0 to 10000 mV
40122	121	0x79	VALUE OF ANALOG OUTPUT 2	Read & Write	
40123	122	0x7A	VALUE OF ANALOG OUTPUT 3	Read & Write	
40124	123	0x7B	VALUE OF ANALOG OUTPUT 4	Read & Write	
40125	124	0x7C	VALUE OF ANALOG OUTPUT 5	Read & Write	
40126	125	0x7D	VALUE OF ANALOG OUTPUT 6	Read & Write	
40141	140	0x8C	WATCHDOG TIME	Read & Write Memory	Time in second before watchdog reset in case no transmission. A 0 value disables Watchdog. The default value is 0 s.
40143	142	0x8E	DEFAULT STATE OF DIGITAL OUTPUTS	Read & Write Memory	State of Digital Outputs assigned at the start of the module and watchdog reset. The default value is 0.

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description
40144	143	0x8F	DEFAULT STATE OF ANALOG OUTPUTS (DIGITAL)	Read & Write Memory	State of Analog Outputs assigned at the start of the module and watchdog reset. The default value is 0.
40145	144	0x90	DEFAULT STATE OF ANALOG OUTPUT 1	Read & Write Memory	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.
40146	145	0x91	DEFAULT STATE OF ANALOG OUTPUT 2	Read & Write Memory	
40147	146	0x92	DEFAULT STATE OF ANALOG OUTPUT 3	Read & Write Memory	
40148	147	0x93	DEFAULT STATE OF ANALOG OUTPUT 4	Read & Write Memory	
40149	148	0x94	DEFAULT STATE OF ANALOG OUTPUT 5	Read & Write Memory	
40150	149	0x95	DEFAULT STATE OF ANALOG OUTPUT 6	Read & Write Memory	
40151	150	0x96	UNIVERSAL INPUT 1 CONFIGURATION	Read & Write Memory	
40152	151	0x97	UNIVERSAL INPUT 2 CONFIGURATION	Read & Write Memory	
40153	152	0x98	UNIVERSAL INPUT 3 CONFIGURATION	Read & Write Memory	
40154	153	0x99	UNIVERSAL INPUT 4	Read &	

Value	Description / Sensor
0	Resistance measurement off
1	10K3A1 NTC
2	10K4A1 NTC

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description
			CONFIGURATION	Write Memory	3 10K NTC Carel
40155	154	0x9A	UNIVERSAL INPUT 5 CONFIGURATION	Read & Write Memory	4 20K6A1 NTC
					5 2,2K3A1 NTC B=3975K
40156	155	0x9B	UNIVERSAL INPUT 6 CONFIGURATION	Read & Write Memory	6 3K3A1 NTC
					7 30K6A1 NTC
40157	156	0x9C	UNIVERSAL INPUT 7 CONFIGURATION	Read & Write Memory	8 SIE1
					9 TAC1
40158	157	0x9D	UNIVERSAL INPUT 8 CONFIGURATION	Read & Write Memory	10 SAT1
					16 Pt1000
					17 Ni1000
					+128 Voltage measurement off
40159	158	0x9E	FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 1	Read & Write Memory	<p>Filter time constant, expressed in seconds in the range from 0 to 60 seconds.</p> <p>A value of 0 disables the filter.</p> <p>The default value is 2s.</p> <p>Filter time constant, expressed in seconds in the range from 0 to 60 seconds.</p> <p>A value of 0 disables the filter.</p> <p>The default value is 2s.</p>
40160	159	0x9F	FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 2	Read & Write Memory	
40161	160	0xA0	FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 3	Read & Write Memory	
40162	161	0xA1	FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 4	Read & Write Memory	
40163	162	0xA2	FILTER TIME CONSTANT	Read &	

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description										
			OF THE UNIVERSAL INPUT 5	Write Memory											
40164	163	0xA3	FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 6	Read & Write Memory											
40165	164	0xA4	FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 7	Read & Write Memory											
40166	165	0xA5	FILTER TIME CONSTANT OF THE UNIVERSAL INPUT 8	Read & Write Memory											
40167	166	0xA6	RESOLUTION OF THE UNIVERSAL INPUTS	Read & Write Memory	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.										
40168	167	0xA7	ANALOG OUTPUT 1 CONFIGURATION	Read & Write Memory	By default, all measurements are done with 12-bit resolution.										
40169	168	0xA8	ANALOG OUTPUT 2 CONFIGURATION	Read & Write Memory	Configuring the mode of Analog Output according to the following table:										
40170	169	0xA9	ANALOG OUTPUT 3 CONFIGURATION	Read & Write Memory	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table>	Value	Description	0 (default)	Voltage output 0-10V	1	PWM 1Hz	2	PWM 10Hz	3	PWM 100Hz
Value	Description														
0 (default)	Voltage output 0-10V														
1	PWM 1Hz														
2	PWM 10Hz														
3	PWM 100Hz														
40171	170	0xAA	ANALOG OUTPUT 4 CONFIGURATION	Read & Write Memory											
40172	171	0xAB	ANALOG OUTPUT 5 CONFIGURATION	Read &											

Modbus Address	Dec Addr	Hex Adr.	Register name	Access	Description				
				Write Memory	<table border="1"> <tr> <td>4</td> <td>PWM 0.1Hz</td> </tr> <tr> <td>5</td> <td>PWM 0.01Hz</td> </tr> </table>	4	PWM 0.1Hz	5	PWM 0.01Hz
4	PWM 0.1Hz								
5	PWM 0.01Hz								
40173	172	0xAC	ANALOG OUTPUT 6 CONFIGURATION	Read & Write Memory					

Table 2. Modbus registers