Guide for configuring and working with the Modbus master (AxACFA) integration module

ACFA PSIM 1.5

Last update 25/07/2025

Table of Contents

1 List of terms used in the Guide for configuring and working with the Modbus master (AxACFA) integration module

Modbus is a communication protocol, which is based on the master-slave architecture. To transfer the data, it uses the interfaces such as RS-485, RS-422, RS-232 (Modbus RTU protocol), and also Ethernet of TCP/IP network (Modbus TCP protocol).

Modbus Device is an automation system device (controller, sensor, actuating mechanism) supporting the Modbus protocol.

Modbus Register is a data type of the Modbus protocol.

2 Introduction into the Guide for configuring and working with the Modbus master (AxACFA) integration module

On the page:

2.1 Purpose of the document

The Guide for configuring and working with the Modbus master (AxACFA) integration module is a reference and information manual and is intended for configuration specialists and operators of the *Modbus master (AxACFA)* integration module.

The Guide has the following information:

1. General information about the *Modbus master (AxACFA)* integration module.
2. Configuration of the *Modbus master (AxACFA)* integration module.
3. Working with the *Modbus master (AxACFA)* integration module.

2.2 General information about the Modbus master (AxACFA) integration module

The *Modbus master (AxACFA)* integration module can perform data exchange, receive events and send commands via Modbus TCP or Modbus RTU protocols.

3 Supported hardware and licensing of the Modbus master (AxACFA) integration module

Per each register (**ACFASensor**).

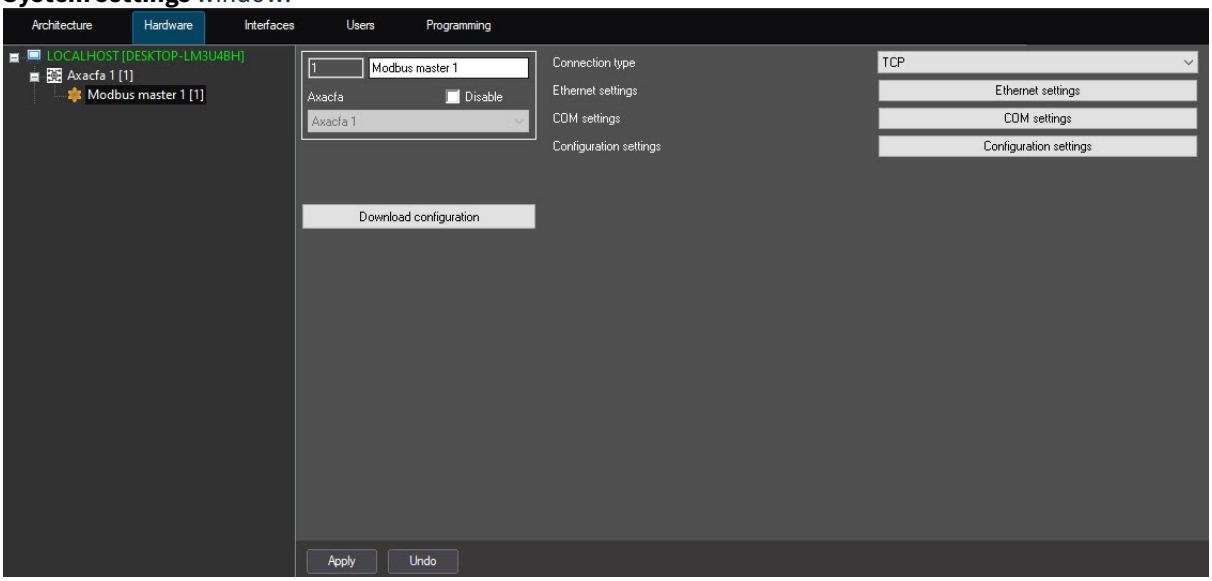
4 Configuring the Modbus master (AxACFA) integration module

4.1 Configuring the parent object of the Modbus master (AxACFA) integration module

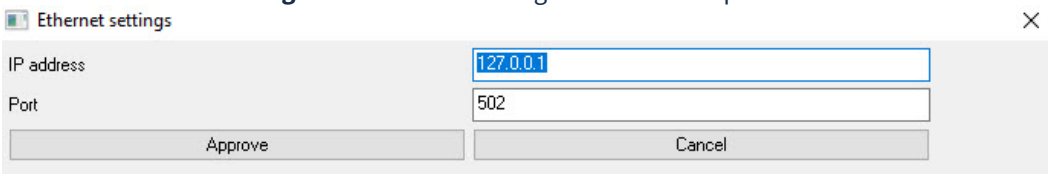
To work with the *Modbus master (AxACFA)* integration module, you must install and configure the *AxACFA* feature. For more details, see Connecting and configuring the AxACFA feature.

To configure the parent object of the *Modbus master (AxACFA)* integration module, do the following:

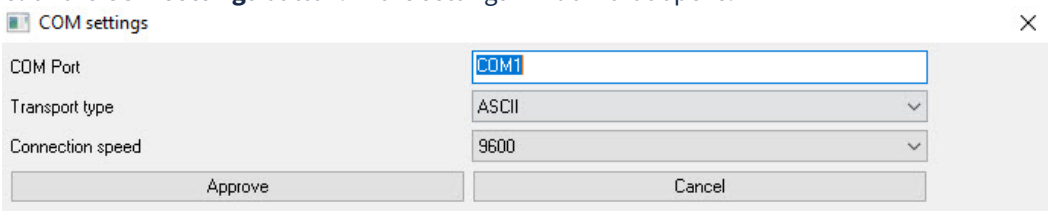
1. Create the **Modbus master** parent object on the basis of the **Axacfa** object on the **Hardware** tab of the **System settings** window.



2. From the **Connection type** drop-down list, select a connection option:
   a. **TCP** or **UDP**:
      i. Click the **Ethernet settings** button. In the settings window that opens:

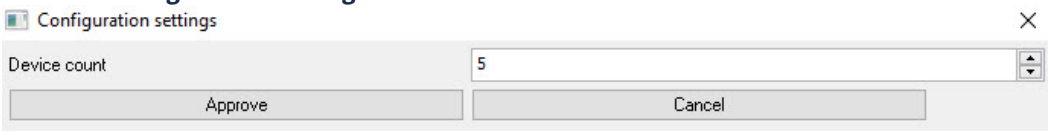


      ii. In the **IP address** field, specify the IP address of the Modbus Server.
      iii. In the **Port** field, specify the port for connection to the Modbus Server.
      iv. Click the **Approve** button to save the settings. Click the **Cancel** button to cancel the changes.
      v. Click the **Apply** button to apply the changes.
   b. **COM Port**:
      i. Click the **COM settings** button. In the settings window that opens:



ii. In the **COM Port** field, specify the number of the COM port for connection to the Modbus Server.
iii. From the **Transport type** drop-down list, select the **ASCII** or **RTU** protocol type for connection to the Modbus Server.
iv. From the **Connection speed** drop-down list, select the COM port bit transfer rate per second: 9600, 19200, 57600, 115200.
v. Click the **Approve** button to save the settings. Click the **Cancel** button to cancel the changes.
vi. Click the **Apply** button to apply the changes.

3. Create child objects by autogeneration or manually:
   a. To autogenerate objects:
      i. Click the **Configuration settings** button.



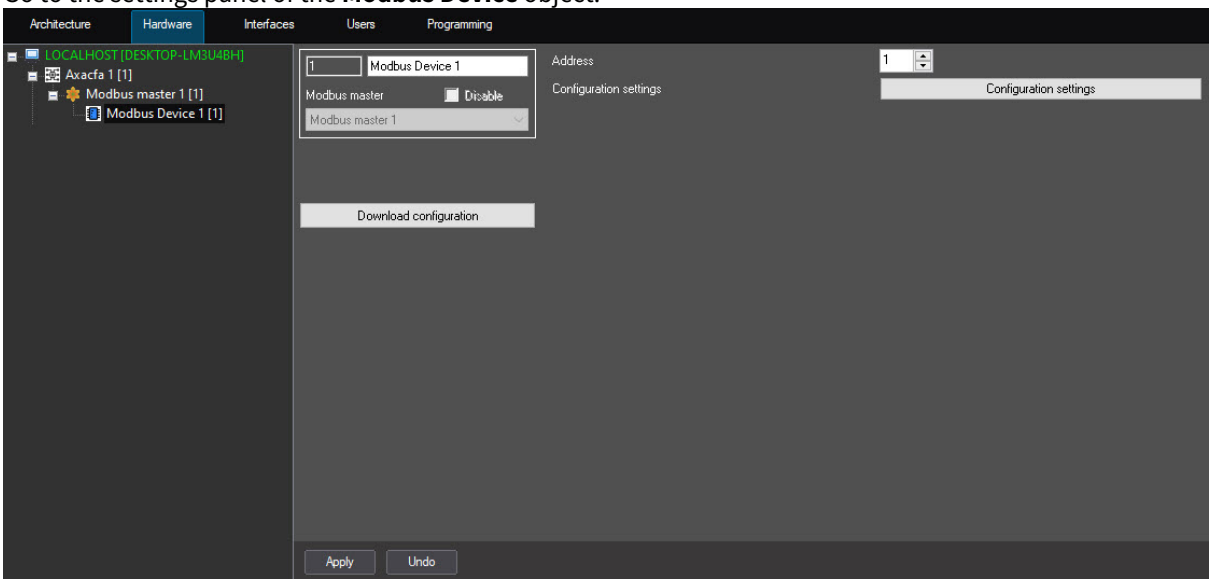
      ii. In the **Device count** field, specify the number of devices connected via the Modbus protocol.
      iii. Click the **Approve** button to save the settings. Click the **Cancel** button to cancel the changes.
      iv. Click the **Download configuration** button to download the configuration and build the hardware tree.
   b. Create child objects on the basis of the parent object manually.

Configuration of the parent object of the *Modbus master (AxACFA)* integration module is complete.

4.2 Configuring the device of the Modbus master (AxACFA) integration module

To configure the Modbus device created on the basis of the parent object of the *Modbus master (AxACFA)* integration module by autogeneration or manually, do the following:

1. Go to the settings panel of the **Modbus Device** object.

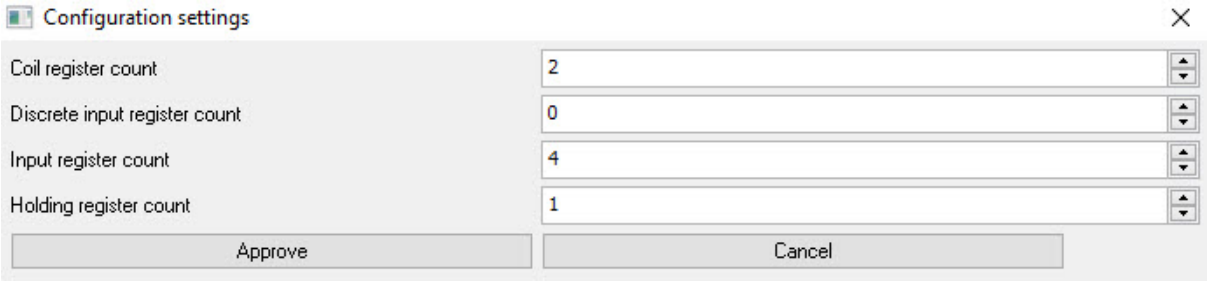


2. In the **Address** field, specify the address of the device connected via the Modbus protocol.

> **Note**

For some devices, which are connected via the Modbus protocol, you must forcibly specify the address that exceeds the current value by 1. This depends on the settings of the device (see the manufacturer's official documentation).

3. Click the **Configuration settings** button to set up the configuration.



   a. In the configuration settings window that opens, specify the number of registers of the required type in the corresponding fields.
   b. Click the **Approve** button to save the settings. Click the **Cancel** button to cancel the changes.
4. Click the **Download configuration** button to download the configuration of this device connected via the Modbus protocol.
5. Click the **Apply** button to apply the settings.

Configuration of the device of the *Modbus master (AxACFA)* integration module is complete.

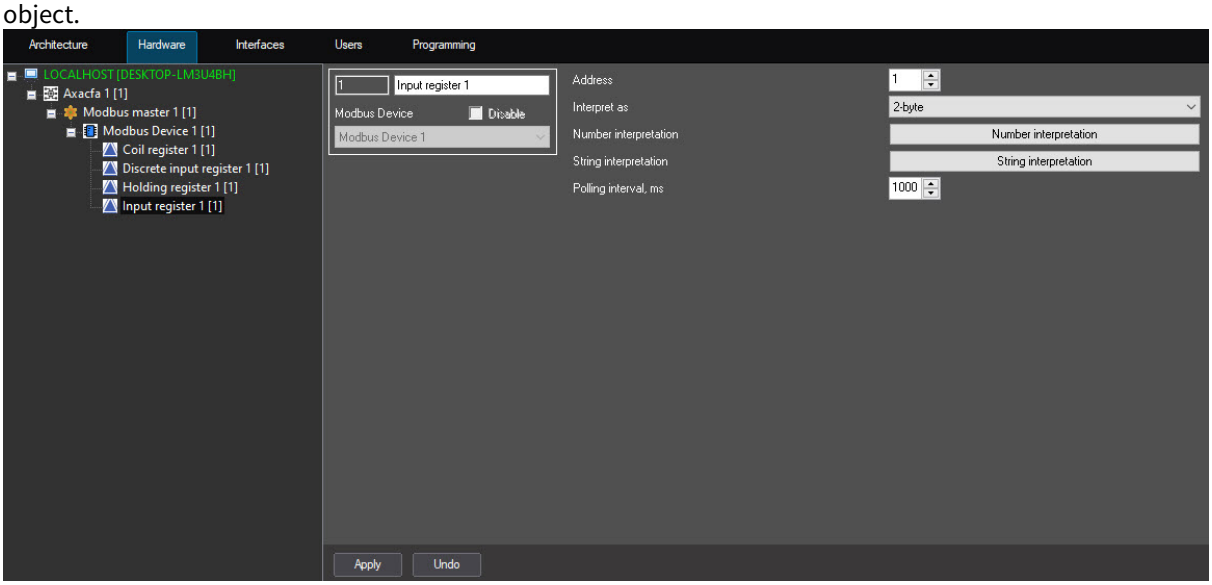
4.3 Configuring the input register and holding register of the Modbus master (AxACFA) integration module

Note

The input register of the *Modbus master (AxACFA)* integration module refers to an Analog Input. An Analog Input can only be read by receiving the state of this input.
The holding register of the *Modbus master (AxACFA)* integration module refers to an Analog Output or a register that stores some values that can be both written and read.
Configuration is shown with the input register as an example. You can configure the holding register in the same way.

To configure the input register of the *Modbus master (AxACFA)* integration module, do the following:

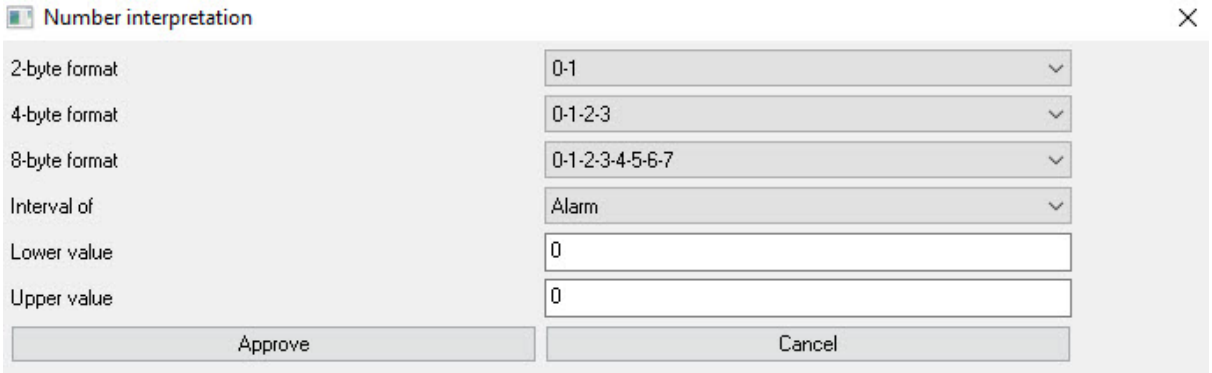
1. Go to the settings panel of the **Input register** object that is created on the basis of the **Modbus Device** object.



2. In the **Address** field, specify the address of the register in the register map of this device.

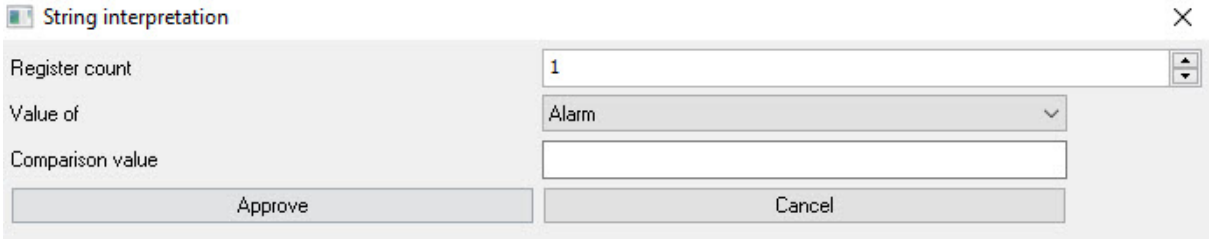
> **Attention!**
>
> - Register map is provided by the manufacturer.
> - Register address is always less than the register number by 1. Register address is specified in the settings.

3. From the **Interpret as** drop-down list, select the data format of the register: **2-byte**, **2-byte unsigned**, **4-byte**, **4-byte unsigned**, **4-byte floating point**, **8-byte**, **8-byte unsigned**, **8-byte double precision**, **String**.
4. In the **Polling interval, ms** field, specify in milliseconds the period of polling and reading new data from the register of the Modbus device.
5. Click the **Number interpretation** button to configure the numerical format of the data in the window that opens.



   a. From the **2-byte format** drop-down list, select a bitmask for 2 bytes.
   b. From the **4-byte format** drop-down list, select a bitmask for 4 bytes.
   c. From the **8-byte format** drop-down list, select a bitmask for 8 bytes.
   d. From the **Interval of** drop-down list, select **Alarm** (default) or **Normal**.
   e. In the **Lower value** field, specify the lower value of the alarm or normal values.
   f. In the **Upper value** field, specify the upper value of the alarm or normal values.
   g. Click the **Approve** button to save the settings. Click the **Cancel** button to cancel the changes.

6. Click the **String interpretation** button to configure the string format of the data in the window that opens.



   a. In the **Register count** field, specify the number of the used registers. The default value is **1**.
   b. From the **Value of** drop-down list, select **Alarm** (default) or **Normal**.
   c. In the **Comparison value** field, specify the value with which the string is compared.
   d. Click the **Approve** button to save the settings. Click the **Cancel** button to cancel the changes.
7. Click the **Apply** button to apply the settings.

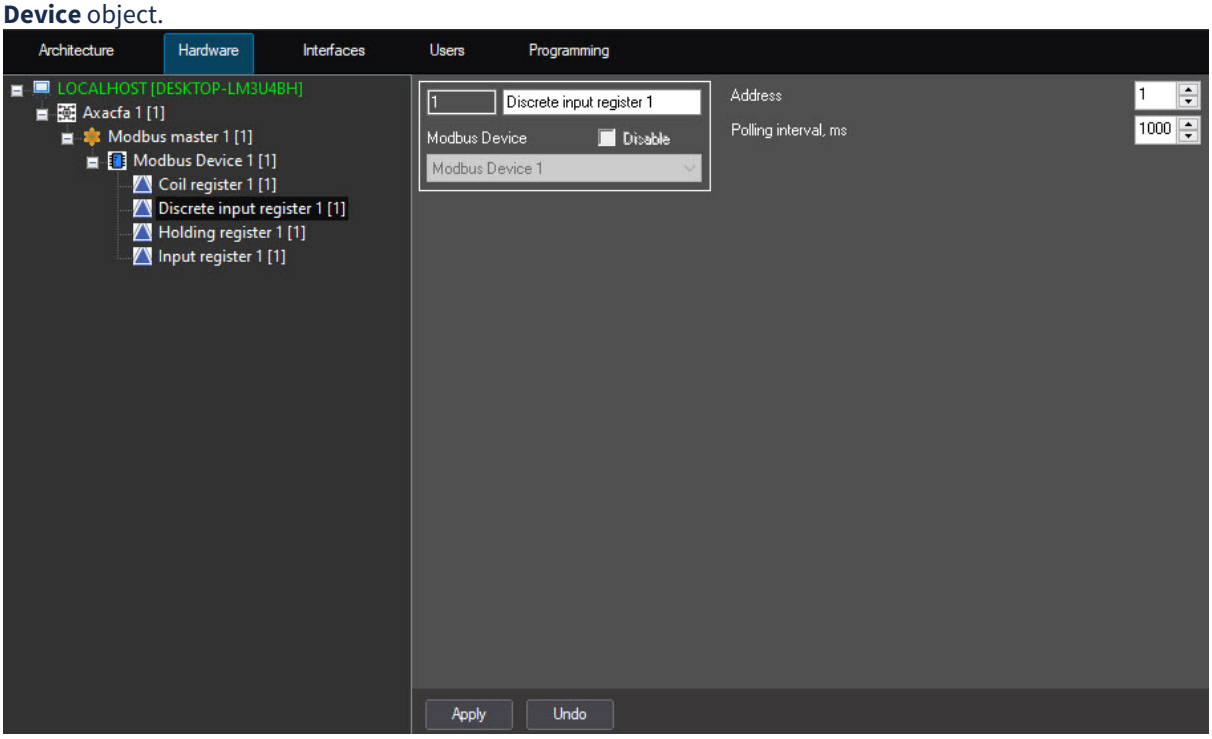
Configuration of the input register and holding register of the *Modbus master (AxACFA)* integration module is complete.

4.4 Configuring the coil register and discrete input register of the Modbus master (AxACFA) integration module

> **Note**
>
> The coil register of the *Modbus master (AxACFA)* integration module refers to a Digital Output or Coil. The coil register can be both read and written.
> The discrete input register of the *Modbus master (AxACFA)* integration module refers to a Digital Input. The discrete input register can only be read by receiving the actual state of this input on a sensor or device. Configuration is shown with the discrete input register as an example. You can configure the coil register in the same way.

To configure the discrete input register of the *Modbus master (AxACFA)* integration module, do the following:

1. Go to the settings panel of the **Discrete input register** object that is created on the basis of the **Modbus Device** object.



2. In the **Address** field, specify the address of the register in the register map of this device.

> **Attention!**
>
> - Register map is provided by the manufacturer.
> - Register address is always less than the register number by 1. Register address is specified in the settings.

3. In the **Polling interval, ms** field, specify in milliseconds the period of polling and reading new data from the register of the Modbus device.
4. Click the **Apply** button to apply the settings.

Configuration of the coil register and discrete input register of the *Modbus master (AxACFA)* integration module is complete.

5 Working with the Modbus master (AxACFA) integration module

5.1 General information on working with the Modbus master (AxACFA) integration module

The following interface objects are used to work with the *Modbus master (AxACFA)* integration module:

1. **Map**.
2. **Event viewer**.

For the information on configuring these interface objects, see *Axxon PSIM* Administrator's Guide.

For the information on using these interface objects, see *Axxon PSIM* Operator's Guide.

5.2 Managing the parent object of the Modbus master (AxACFA) integration module

You cannot manage the parent object of the *Modbus master (AxACFA)* integration module in the **Map** window.

The parent object of the *Modbus master (AxACFA)* integration module can have the following states:

	Connected
	Disconnected

5.3 Managing the Modbus device of the Modbus master (AxACFA) integration module

You cannot manage the Modbus device of the *Modbus master (AxACFA)* integration module in the **Map** window.

The Modbus device can have the following states:

	Connected
	Disconnected

5.4 Managing the discrete input register of the Modbus master (AxACFA) integration module

You cannot manage the discrete input register of the *Modbus master (AxACFA)* integration module in the **Map** window.

The discrete input register of the *Modbus master (AxACFA)* integration module can have the following states:

	Enabled
	Disabled
	Unknown

5.5 Managing the input register of the Modbus master (AxACFA) integration module

You can manage the input register of the *Modbus master (AxACFA)* integration module in the **Map** window using the menu of the **Input register** object.

Command to manage the input register of the *Modbus master (AxACFA)* integration module:

- Read—read the current value from the input register.

The input register of the *Modbus master (AxACFA)* integration module can have the following states:

	Normalized
	Alarm
	Unknown

5.6 Managing the coil register of the Modbus master (AxACFA) integration module

You can manage the coil register of the *Modbus master (AxACFA)* integration module in the **Map** window using the menu of the **Coil register** object.

Commands to manage the coil register of the *Modbus master (AxACFA)* integration module are described in the table:

Menu command	**Function**
Enable	Write the **false** parameter
Disable	Write the **true** parameter

The coil register of the *Modbus master (AxACFA)* integration module can have the following states:

	Enable
	Disable
	Unknown

5.7 Managing the holding register of the Modbus master (AxACFA) integration module

You can manage the holding register of the *Modbus master (AxACFA)* integration module in the **Map** window using the menu of the **Holding register** object.

Commands to manage the holding register of the *Modbus master (AxACFA)* integration module are described in the table:

Menu command	Function
Read	Read the current value from the holding register
Write	Write the current value to the holding register

The holding register of the *Modbus master (AxACFA)* integration module can have the following states:

	Normalized
	Alarm
	Unknown

5.8 Example of a configured macro when working with the Modbus master (AxACFA) integration module

Creating and using macros
Examples of macros

When working with the *Modbus master (AxACFA)* integration module, you can configure a macro that will trigger when an event is received from a *Modbus master* device.

Example of a configured macro: