Data exchange between SMI2-M and PR

# Introduction

The example describes how to organize the data exchange between the SMI2-M indicator and the PR programmable relay. In the example we will use PR102, but you can use any other PR in the same way.

A temperature transmitter with a measuring range 0…100 °C and an output signal 4-20 mA is connected to the analog input of the PR102. The value on the input is transmitted to the SMI2-M to be displayed. Besides, the color of the display should change depending on the displayed value. The network settings of the devices are shown in Tab. 1.

*Table 1. Device network settings*

|  |  |  |
| --- | --- | --- |
| **Parameter** | **SMI2-M** | **PR102** |
| Interface | RS485 | RS485-1 |
| Protocol | Modbus RTU | |
| Operation mode | Slave | Master |
| Address | 16 | - |
| Baud rate | 115200 | |
| Data bits | 8 | |
| Parity | none | |
| Stop bits | 1 | |

# SMI2-M configuration

To configure SMI2-M we will use the akYtecToolPro configurator software.

Proceed as follows:

1. Connect SMI2-M to the PC over a USB-to-Micro-USB connection cable
2. Start akYtecToolPro
3. In a new project, click the toolbar item *Add devices*2021-01-19 19_53_34-akYtec Tool Pro__ - Project not saved (Fig. 1)
4. In the opened dialog, select the interface *STMicroelectronics Virtual COM Port*. The correct COM port number can be found in Windows Device Manager.
5. *Protocol*: ***akYtec Auto Detection Protocol***
6. *Connection setup*: ***Auto***
7. *Find device* > *Address*: ***1***
8. Click the *Search* button
9. The device found appears on the right panel of the window. Select it and click on the *Add devices* button at the bottom right to add the device to the project.
10. Set the following SMI2-M parameters (Fig. 2):
11. *Device settings* > *Operation mode*: ***SLAVE***
12. *Device settings* > *Modbus common* > *Address in Slave mode (Slave ID)*: ***16***
13. *Device settings* > *Display* > *Data type*: ***REAL***

For further parameters, see SMI2-M user guide.

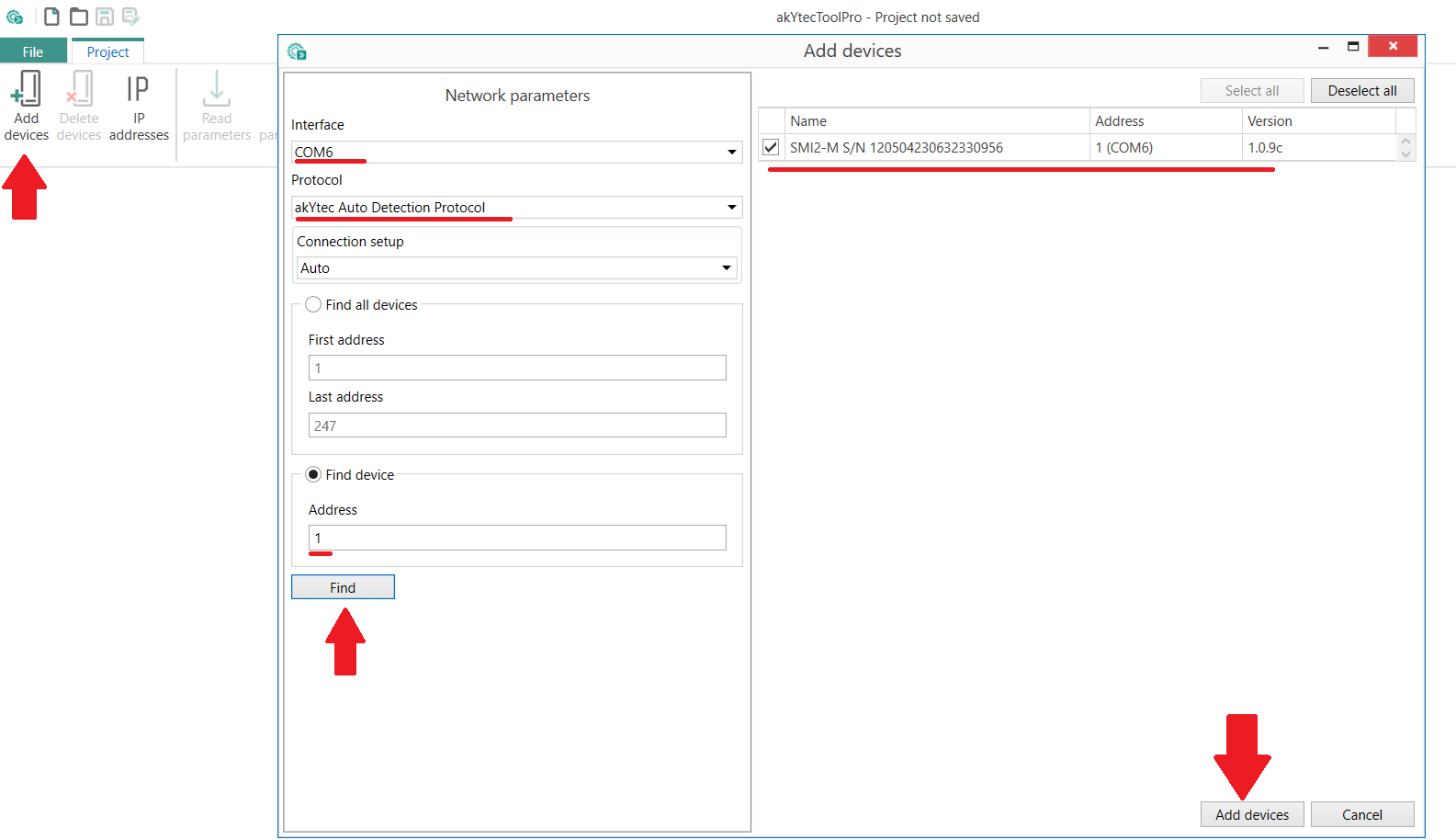


Fig. 1. Add SMI2-M to project

Click *Write parameters* ** toolbar item to write the settings to the device memory.

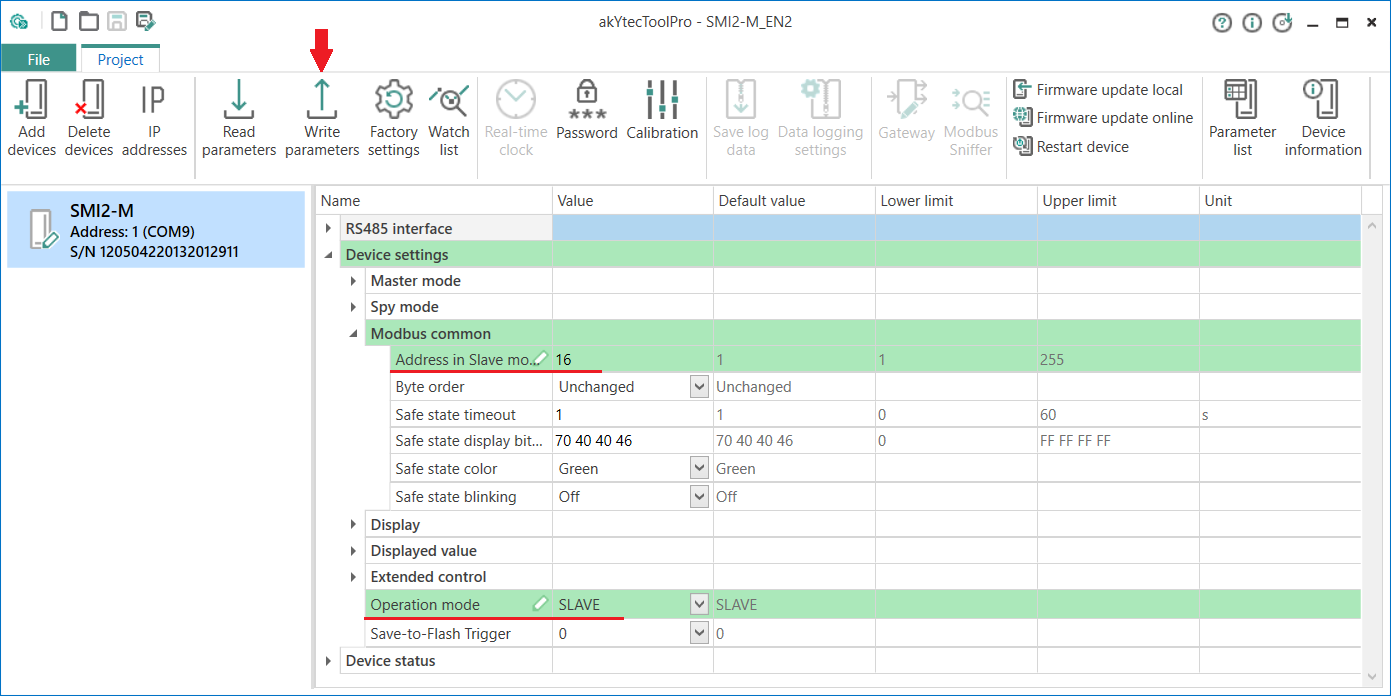


Fig. 2. SMI2-M settings

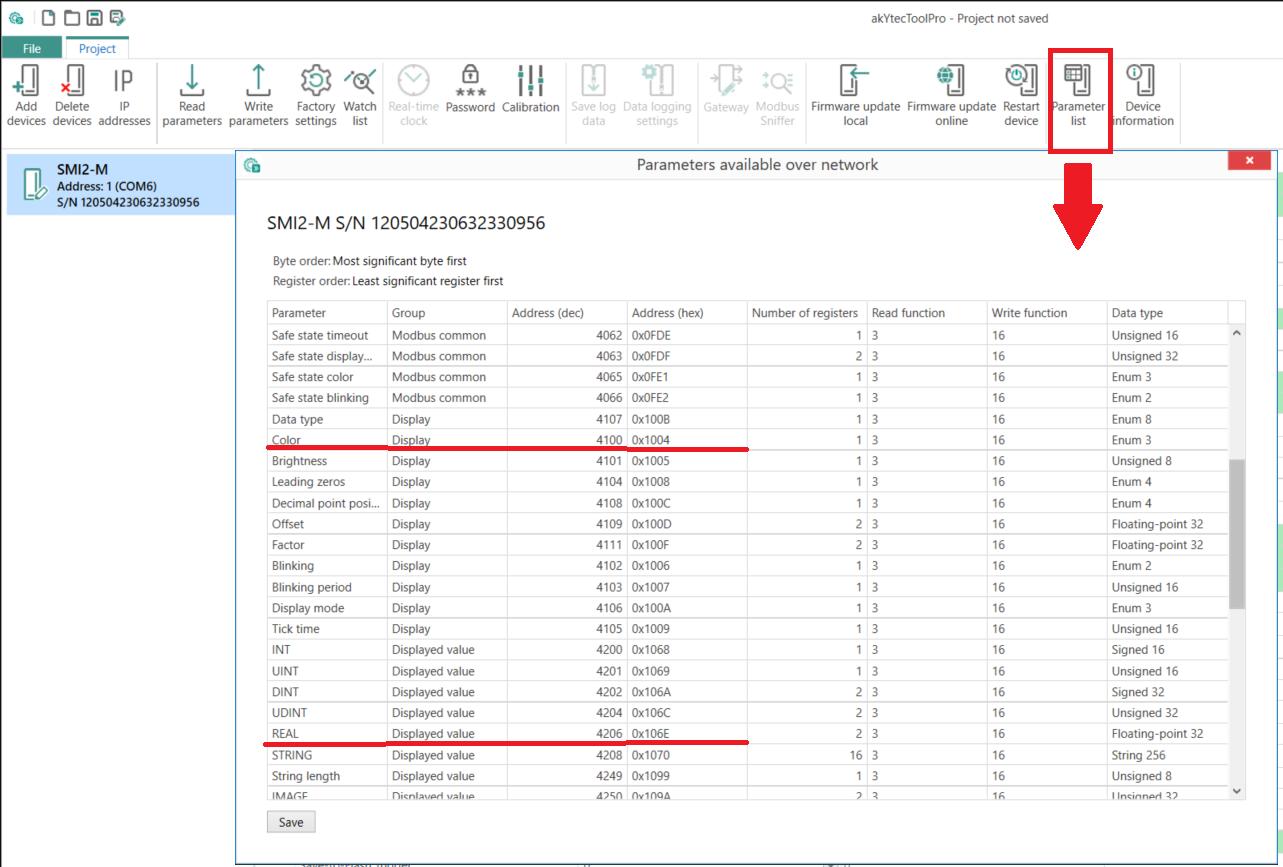


Fig. 3. SMI2-M memory table

Click *Parameter list* toolbar item to open the device memory table (Fig. 3). The displayed REAL value is saved in the register ***4206***, and the display color is saved in the register ***4100***. The Modbus write function for both parametersis ***16***.

# PR102 configuration

Proceed as follows:

1. Create in akYtec ALP a project for the desired PR model
2. Go to menu item *Device > Device configuration*
3. In the opened window *Device configuration*, select the node *Interfaces > RS485* and set the parameters according to Tab. 1 (Fig. 4)
4. Right-click on the *RS485* node and select *Add Slave* (Fig. 5)
5. In the Slave parameters, specify *Address (Slave ID) =****16*** (Tab. 1, Fig. 6)
6. Create two network variables:
   1. ***rValue*** – value to be displayed, REAL, address ***4206***
   2. ***iColor*** – display color, INT, address ***4100***

For register addresses and write functions, see Fig. 3.

For variable ***rValue***, uncheck the option *Write by change* and link the parameter *Start writing* to a BOOL variable ***bWrite*** (Fig. 6, 7).

This prevents the low digits of the variable from flickering on the display due to their rapid change.

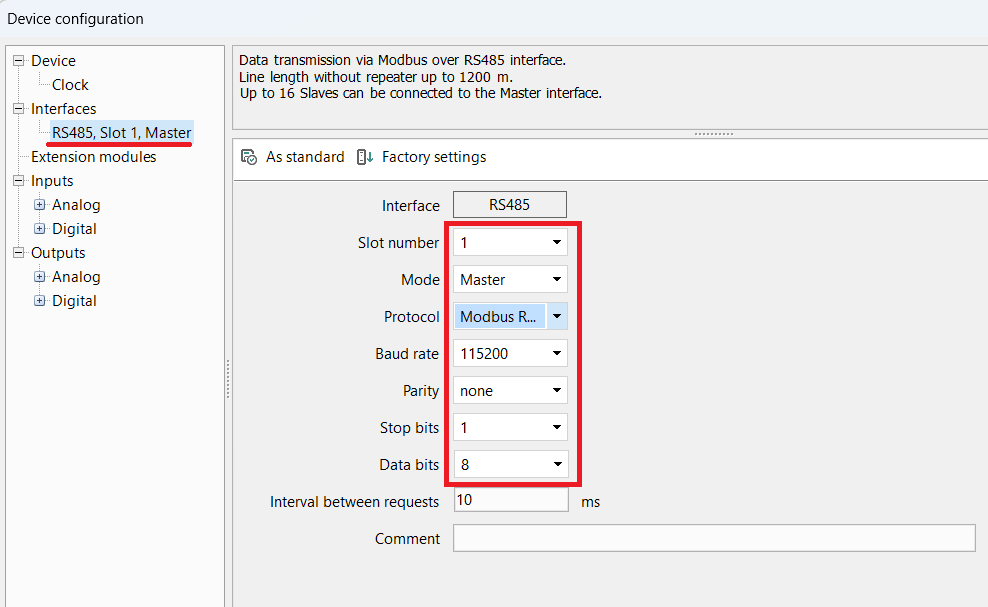


Fig. 4. RS485 settings in Master mode

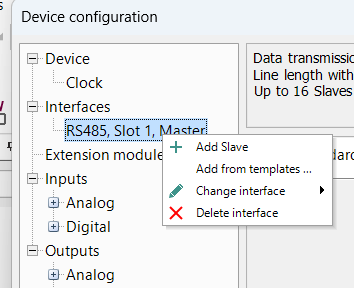


Fig. 5. Add Slave device

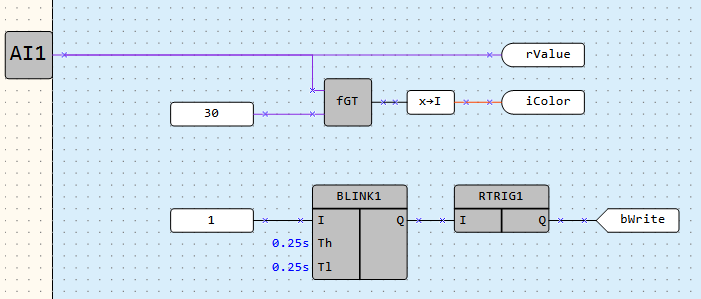


Fig. 6. PR102 program

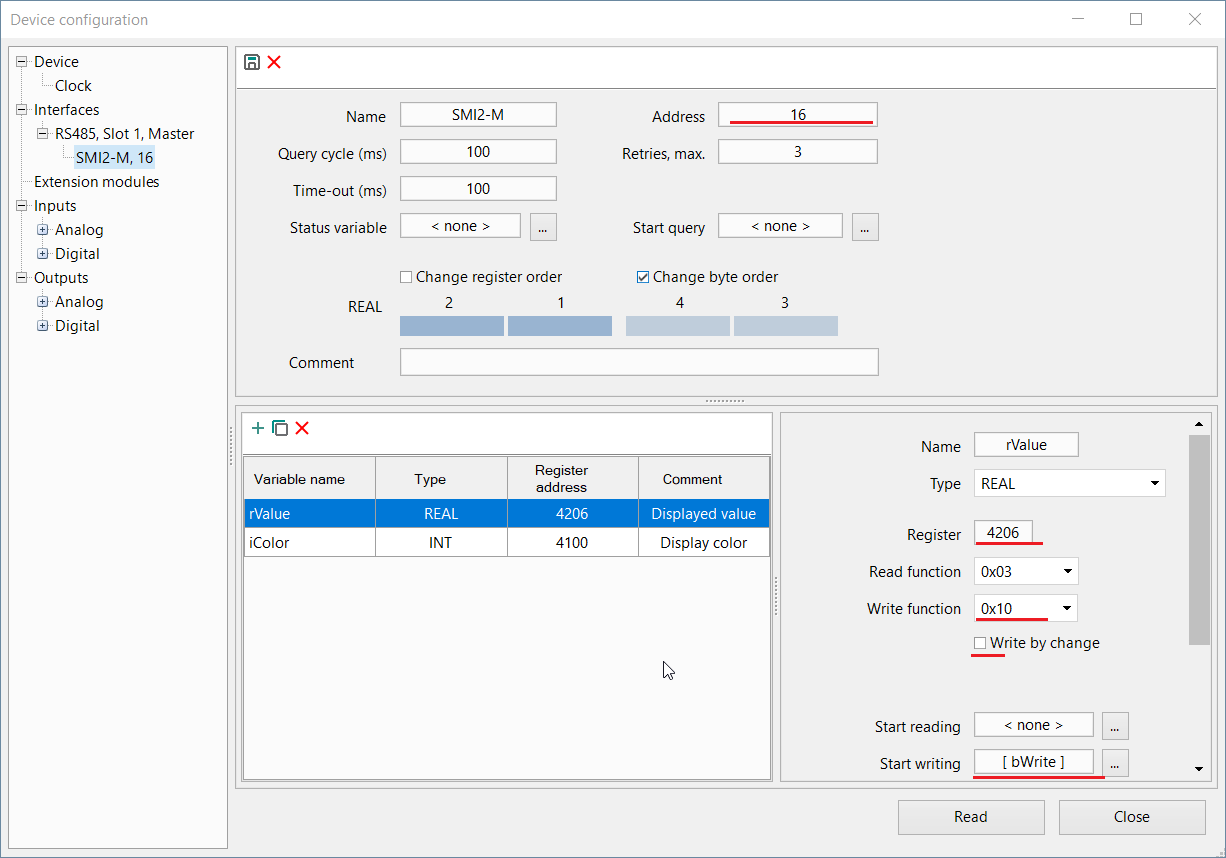


Fig. 7. Variable *rValue*

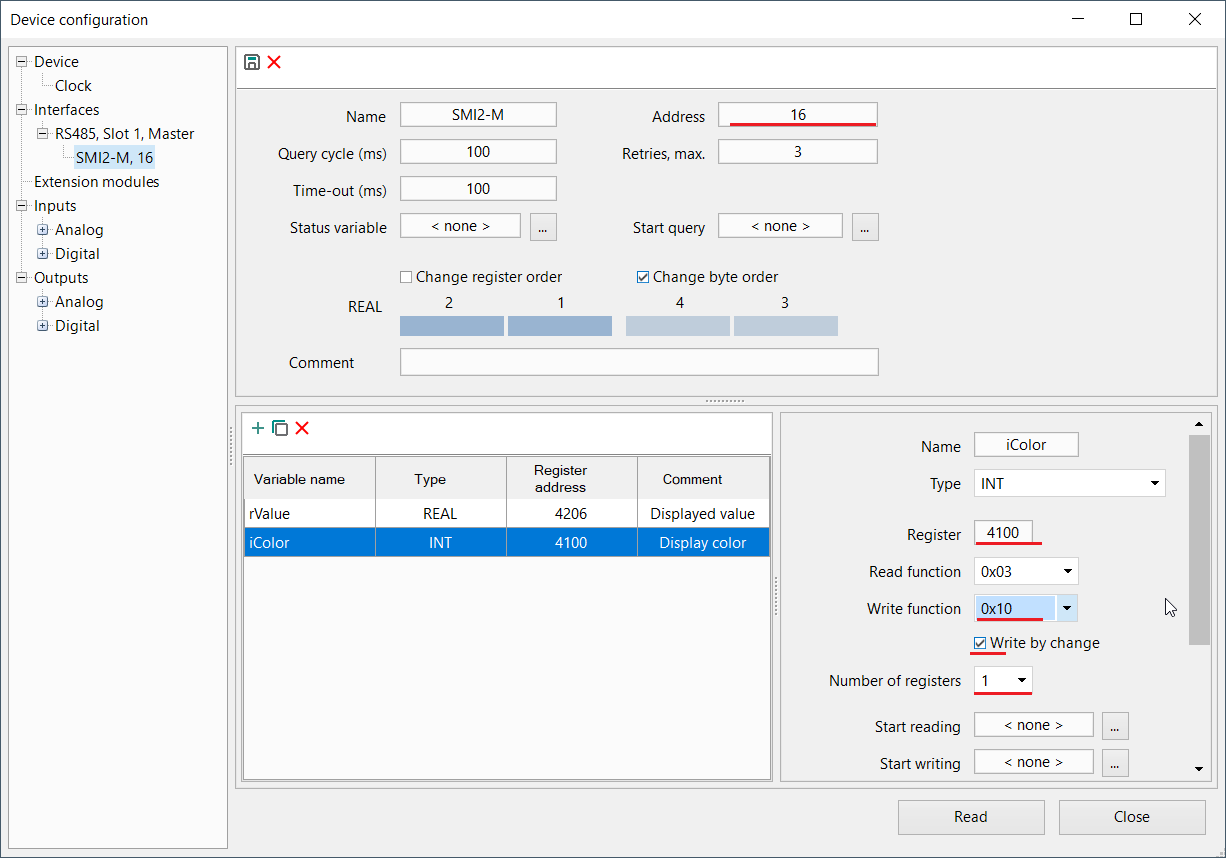


Fig. 8. Variable *iColor*

This problem could be solved by setting the filter time constant of the analog input – but since in practice the value of the displayed variable can be generated in the program or transmitted from devices connected to the second RS485 interface of the PR102 – we use writing as a periodic event in the example. To realize this, we have to assign the ***bWrite*** variable to the *Start writing* parameter (Fig. 6, 7), in which ***1*** is periodically generated.

The variable ***iColor*** is written by change (Fig. 8).

1. Select the input *AI1* in the node *Inputs > Analog* and set its parameters assuming that a temperature transmitter with a measuring range 0…100 °C and an output signal 4-20 mA is connected (Fig. 9).

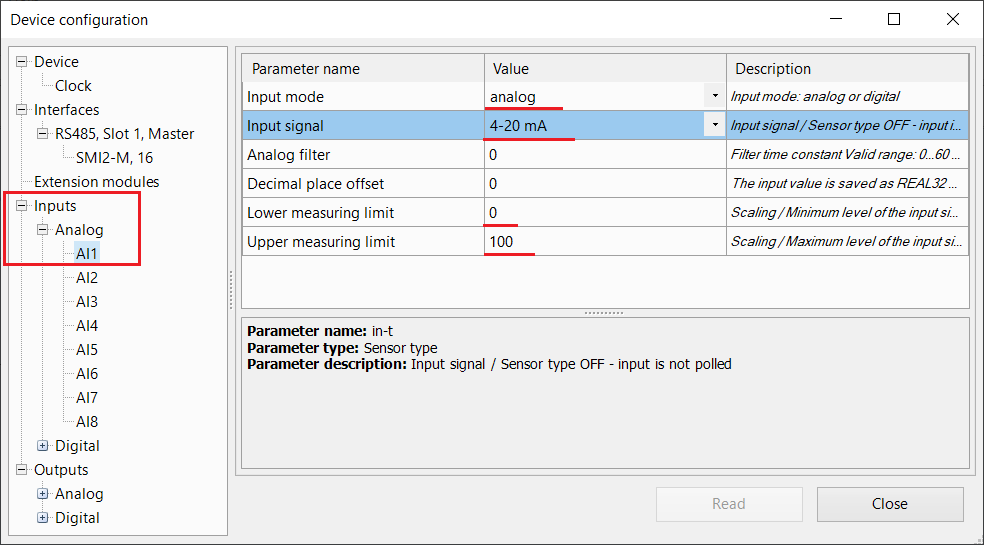


Fig. 9. Analog input settings

The configuration is completed.

# Program

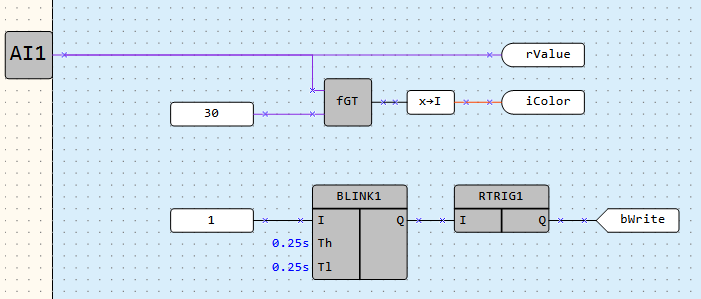


Fig. 10. PR102 program

In the first segment of the program, the value of the ***iColor*** variable is generated. If the measured value is equal or less than 30 °C (after scaling, see Fig. 9), ***iColor*** = ***0*** and the display lights green. If the measured value is greater than 30, ***iColor*** changes to ***1*** and the display color – to red.

The second segment of the program contains a pulse generator for the ***bWrite*** variable, which is used to write the ***rValue*** variable at a fixed time interval (0.25 + 0.25 = 0.5 seconds).