Compatible with
SMI200

Sample Project

SMI200 as Master/Slave in a Modbus Network

Contents

1 Hardware and software

To accurately imitate all steps throughout this sample project, you need to have the following at hand:
- Compact Programmable Controller SMI200
- Programmable Relay PR200
- Anything that can generate a signal of 4-20 mA
- Programming environment akYtec ALP

2 Project objectives

With help of this sample project, you will learn how to:

- Establish communication between SMI200/PR200 and the PC

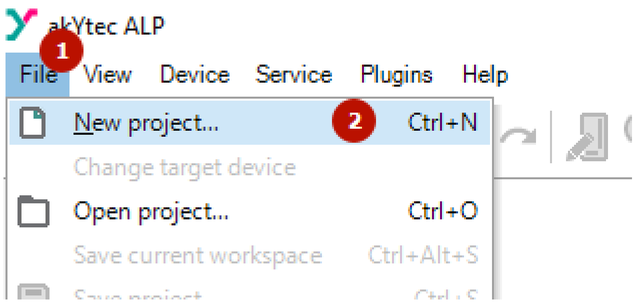
- Configure SMI200 to operate in Slave mode and:
  exchange values with the Master device (PR200) using both an INT variable and a REAL variable

- Configure SMI200 to operate in Master mode and:
  send values from SMI200 to a Slave device (PR200)
  request values from an analog input of this Slave device directly from its Modbus register or using a network variable

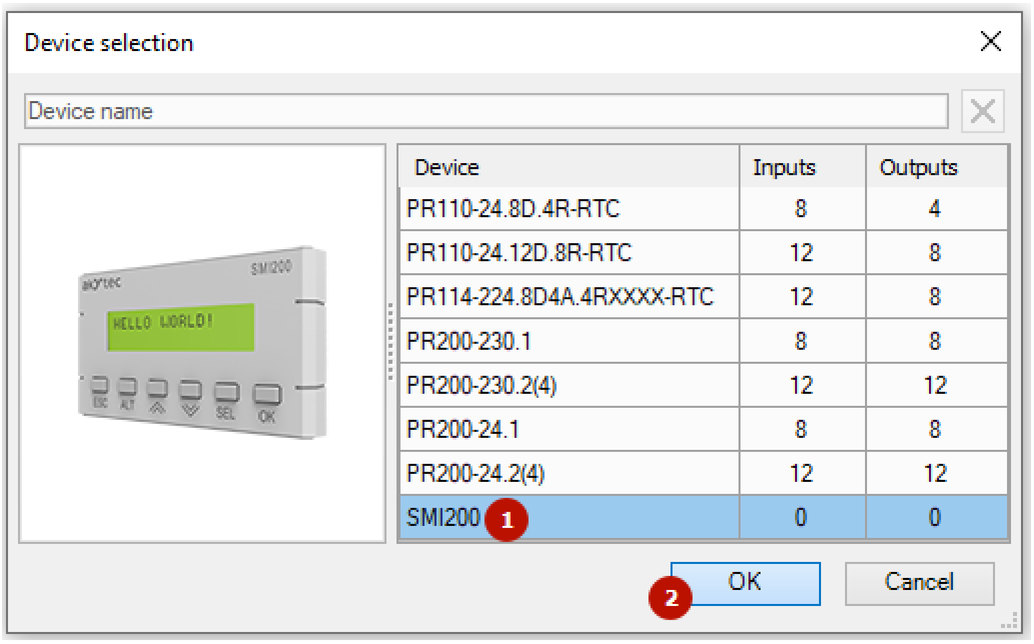
- Create a simple user interface to be displayed on the LCD of SMI200/PR200, thus empowering yourself not only to monitor network values but also to edit them with function keys on the device front

3 Communication between SMI200/PR200 and the PC

Having downloaded, installed, and run akYtec ALP, connect your SMI200 (or PR200 if you are doing Section 4.2) to a USB port of the PC.
In akYtec ALP, in the menu *File* > *New Project*

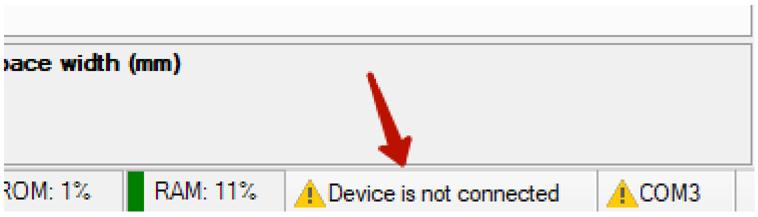


select *SMI200 (or the PR200 of the variant you have at hand)*:

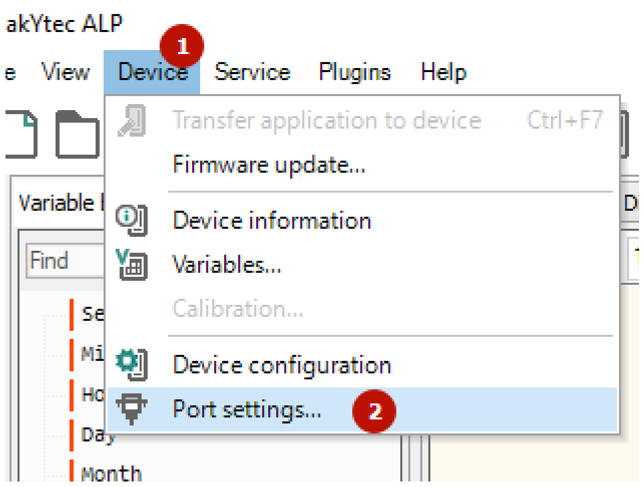


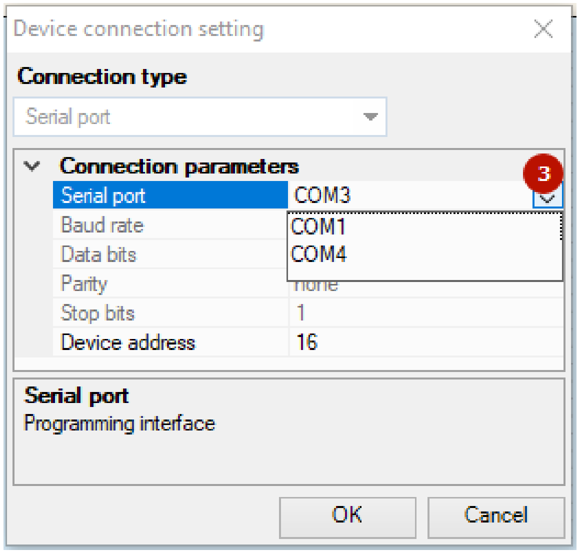
Device	Inputs	Outputs
PR110-24.8D.4R-RTC	8	4
PR110-24.12D.8R-RTC	12	8
PR114-224.8D4A.4RXXXX-RTC	12	8
PR200-230.1	8	8
PR200-230.2(4)	12	12
PR200-24.1	8	8
PR200-24.2(4)	12	12
SMI200	0	0

Check the right lower corner of the *akYtec ALP* window. Most likely, there is something like this:



To overcome this obstacle, you should select the proper serial port in the menu *Device* > *Port settings*:

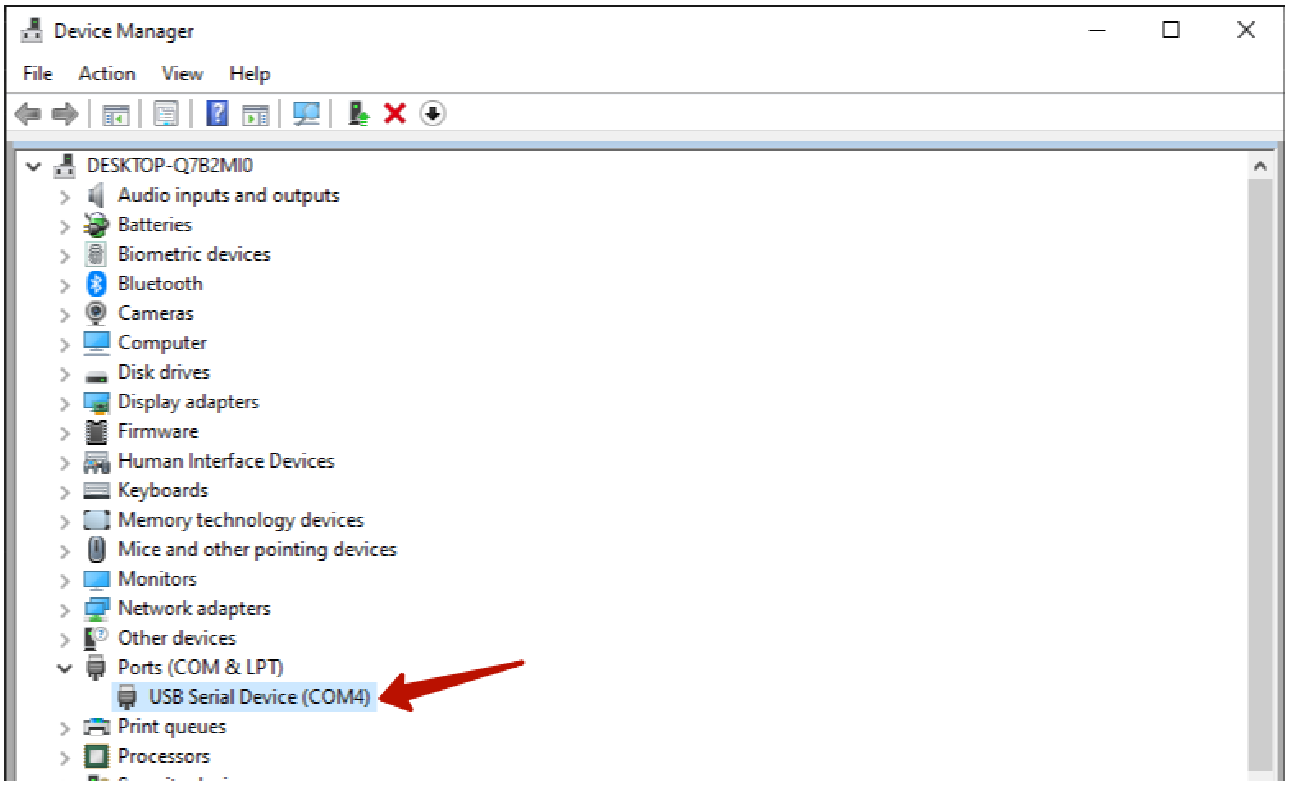




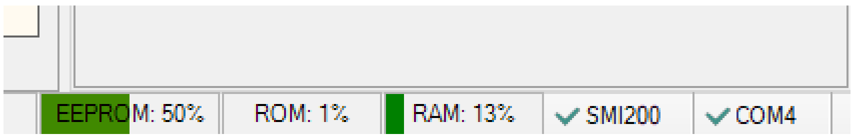
Serial port	COM3
Baud rate	COM1
Data bits	COM4
Parity	none
Stop bits	1
Device address	16

Serial port
Programming interface

The number of the ***Serial port*** in the *Device connection setting* window must agree with the number of the COM port which your SMI200/PR200 has been physically connected to. You can find this number in the device tree of the Windows Device Manager (shortcut *Win+X > Device Manager*) under 'Ports (COM and LPT)'. In the figure below, it is 4; however, your number may differ owing to a different COM ports configuration of your PC.



To ensure you have done everything properly, take another glance at the right lower corner of the *akYtec ALP* window. There should be no exclamation marks:



Compatible with
SMI200

4 Configuration

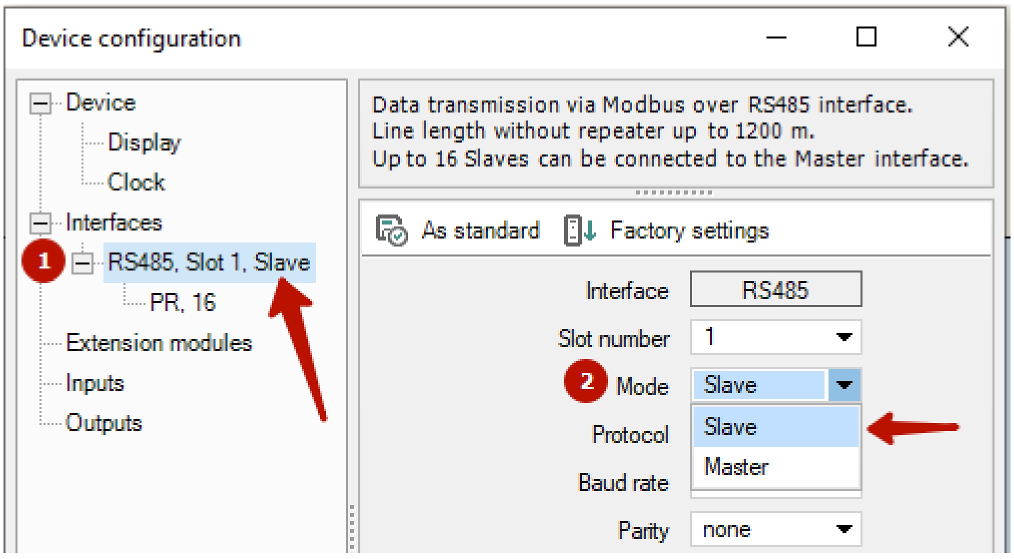
In this sample project, we are going to connect two devices from akYtec – Compact Programmable Controller SMI200 and Programmable Relay PR200 – in a simple Modbus network making them communicate a few values between each other.

There is a control cabinet enclosure with a PR200 snapped on a DIN rail inside. You cannot access the PR200's LCD to check any process values without opening the enclosure door. Nonetheless, one had the forethought to mount a SMI200 at this door, wire it with the PR200, and establish proper communication between them. You could have been that one if you had not stopped reading here.

4.1 Configuration of SMI200 to operate as Slave

STEP 1 Slave identification

Having connected your SMI200 to the PC and established a successful communication between them (see Section 3), open the *Device configuration* dialog by clicking on the *Device configuration* icon in the toolbar and ensure that the (1) device interface is in the Slave mode:

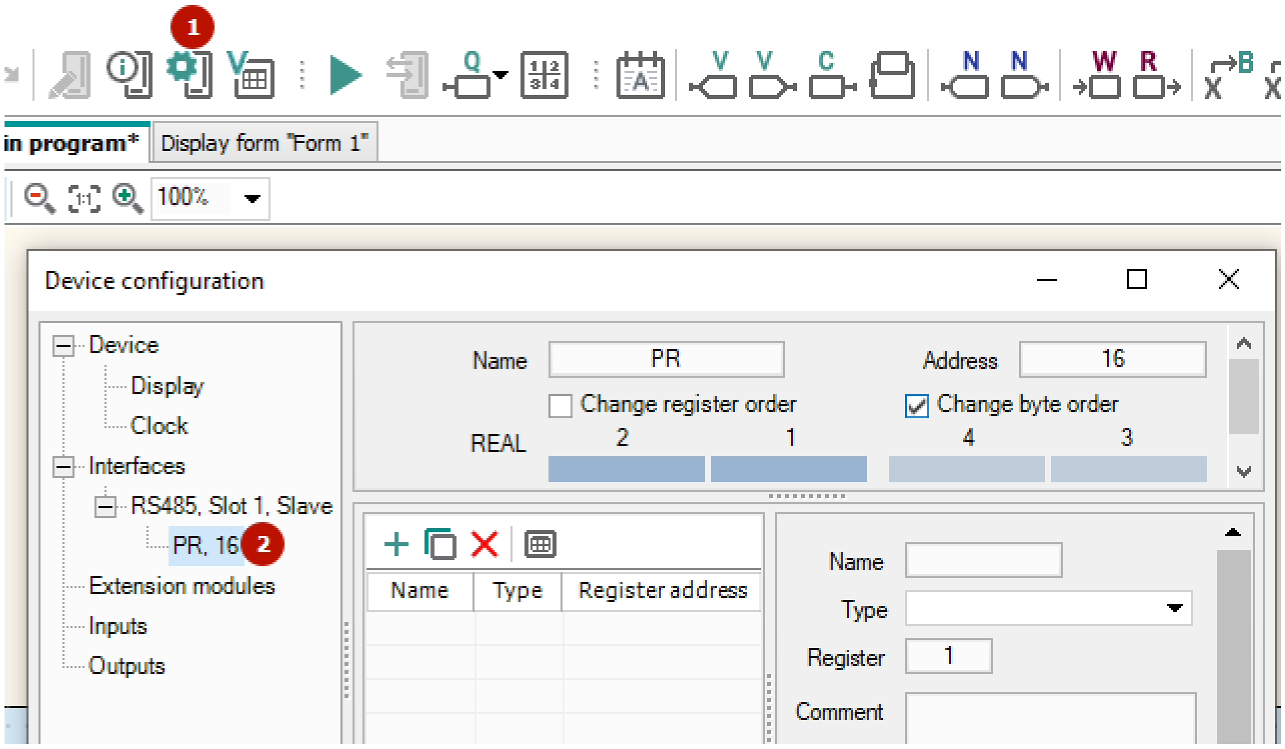


Data transmission via Modbus over RS485 interface.
Line length without repeater up to 1200 m.
Up to 16 Slaves can be connected to the Master interface.

Interface	RS485
Slot number	1
Mode	Slave
Protocol	
Baud rate	
Parity	none

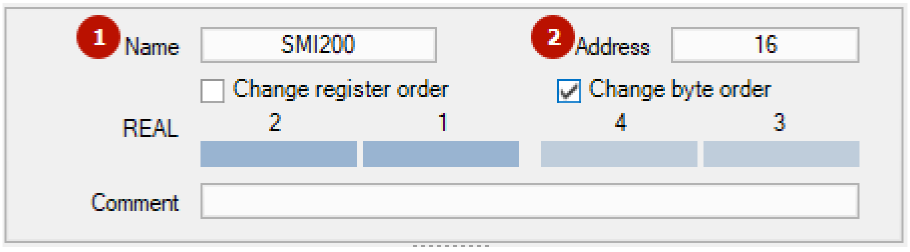
Otherwise, select **Slave** on the (2) *Mode* dropdown.

Select the ***PR, 16*** item in the device tree to the left:



Name: PR
Address: 16
Change register order ☐
Change byte order ✓
REAL 2 1 4 3
Register: 1

Now you can fill in all the essential information which will assist the PR200 (Master) to identify the SMI200 (Slave):

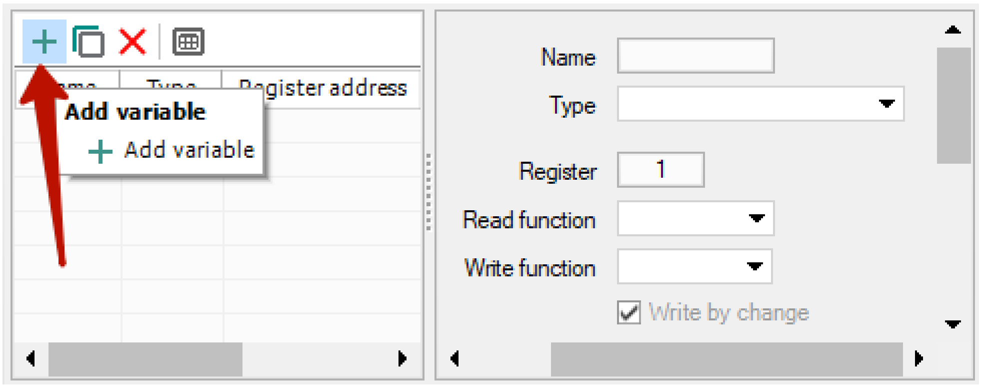


Name: SMI200
Address: 16
Change register order ☐
Change byte order ✓
REAL 2 1 4 3
Comment:

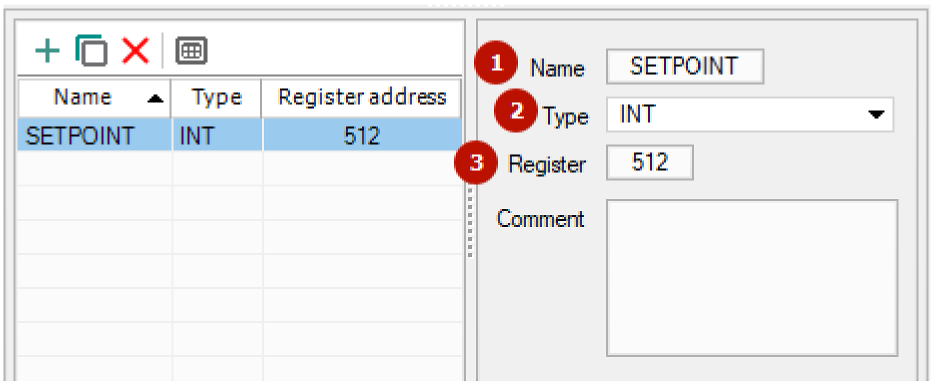
(1) ***Name*** – You can choose any reasonable name to put here. In this case, *SMI200* is reasonable enough.
(2) ***Address*** – The number is *16* because it is the default Modbus Slave address of SMI200. Anyway, you can change it if you like.
The ***Change register order*** and ***Change byte order*** checkboxes remain untouched; anyway, their adjustment will be overridden by the corresponding checkboxes of the Modbus Master which we will configure Section 4.1.2, Step 1.

STEP 2 Declaration of network variables

Also, in the *Device configuration* dialog, you can declare network variables by clicking the + *Add variable* icon:



First, we add one network variable of (1) ***Type*** *INT* with an arbitrary (2) ***Name***, say *SETPOINT*:



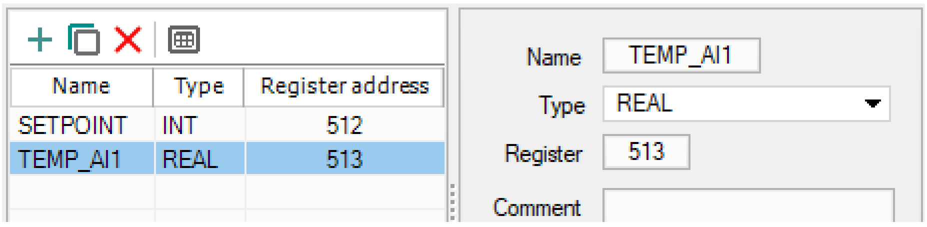
Name	Type	Register address
SETPOINT	INT	512

(1) Name: SETPOINT
(2) Type: INT
(3) Register: 512
Comment:

(3) ***Register*** – This is the address of the register which is to contain the value of the variable SETPOINT. The register address must be defined, otherwise the variable will not be accessible. According to the SMI200 user guide (Table 7.2, page 10), a network variable of type INT must have the register address in the range between 0x0200 and 0x02FF. Since the ***Register*** parameter in this dialog must be entered using the decimal numeral system, you can choose any in the range between 512 and 767 respectively. We have chosen *512*.

NOTE When adding network variables in the *Device configuration* dialog in akYtec ALP, you must define the *Register* parameter using the decimal numeral system

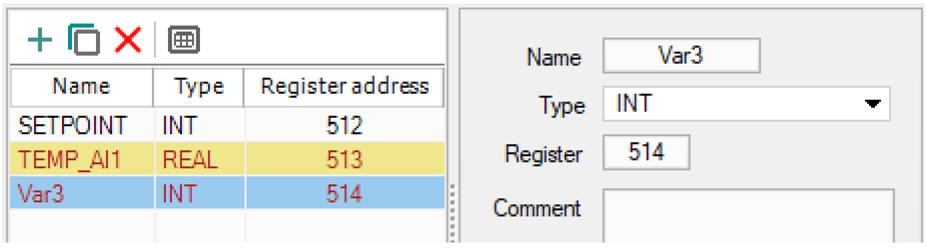
In a very similar manner, we add a REAL variable with an arbitrary name and a vacant address, say *TEMP_AI1* and *513*:



Name	Type	Register address
SETPOINT	INT	512
TEMP_AI1	REAL	513

Name: TEMP_AI1
Type: REAL
Register: 513
Comment:

You might be tempted to add one more variable. Caution, the register address of this variable must not be 514 in this case:



Name	Type	Register address
SETPOINT	INT	512
TEMP_AI1	REAL	513
Var3	INT	514

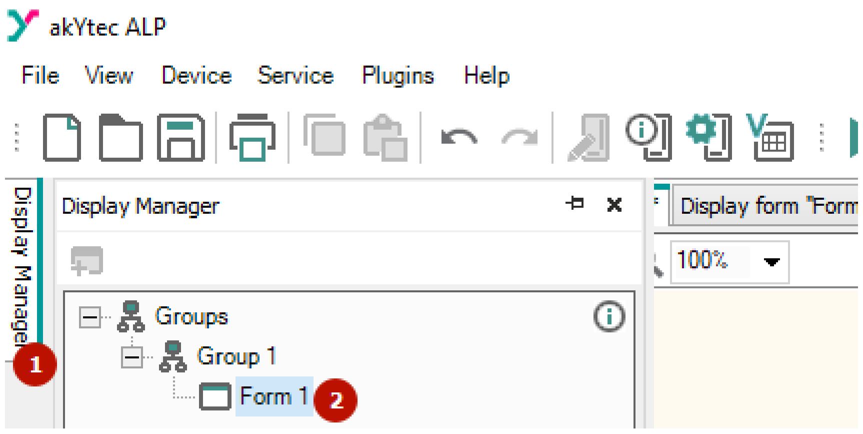
Name: Var3
Type: INT
Register: 514
Comment:

The thing is that one REAL variable takes two registers in a row.

At this point, the SMI200 has already been configured enough to operate as a Modbus Slave device. However, for the sake of illustrative purposes, a few touches wouldn't hurt before transferring the present settings to the device memory. The touches are about creating a user interface on SMI200’s LCD to allow you to monitor the above-declared variables and assign them some values with the function keys on the device front.

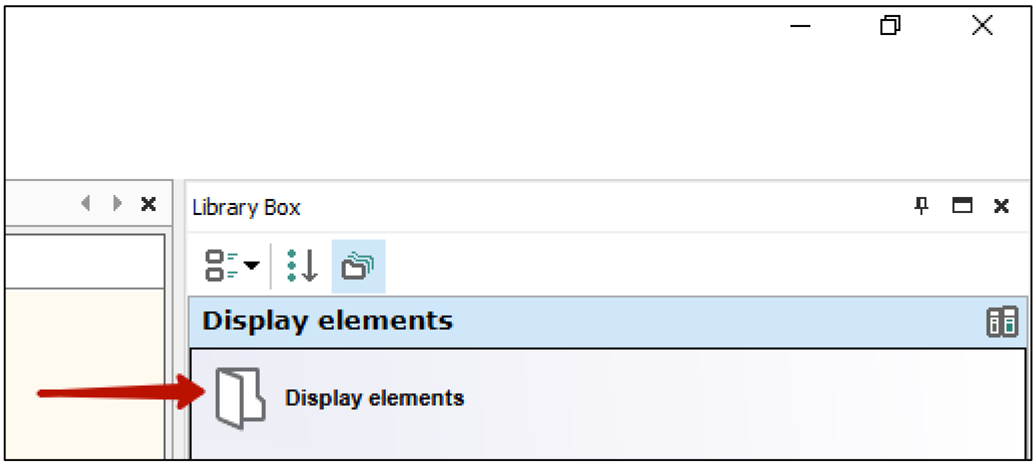
STEP 3 User interface for SMI200/PR200’s LCD

Hover the mouse pointer over the (1) *Display Manager* tab at the top-left corner of the *akYtec ALP* window and double-click on the (2) *Form 1* item in the resulting tree:

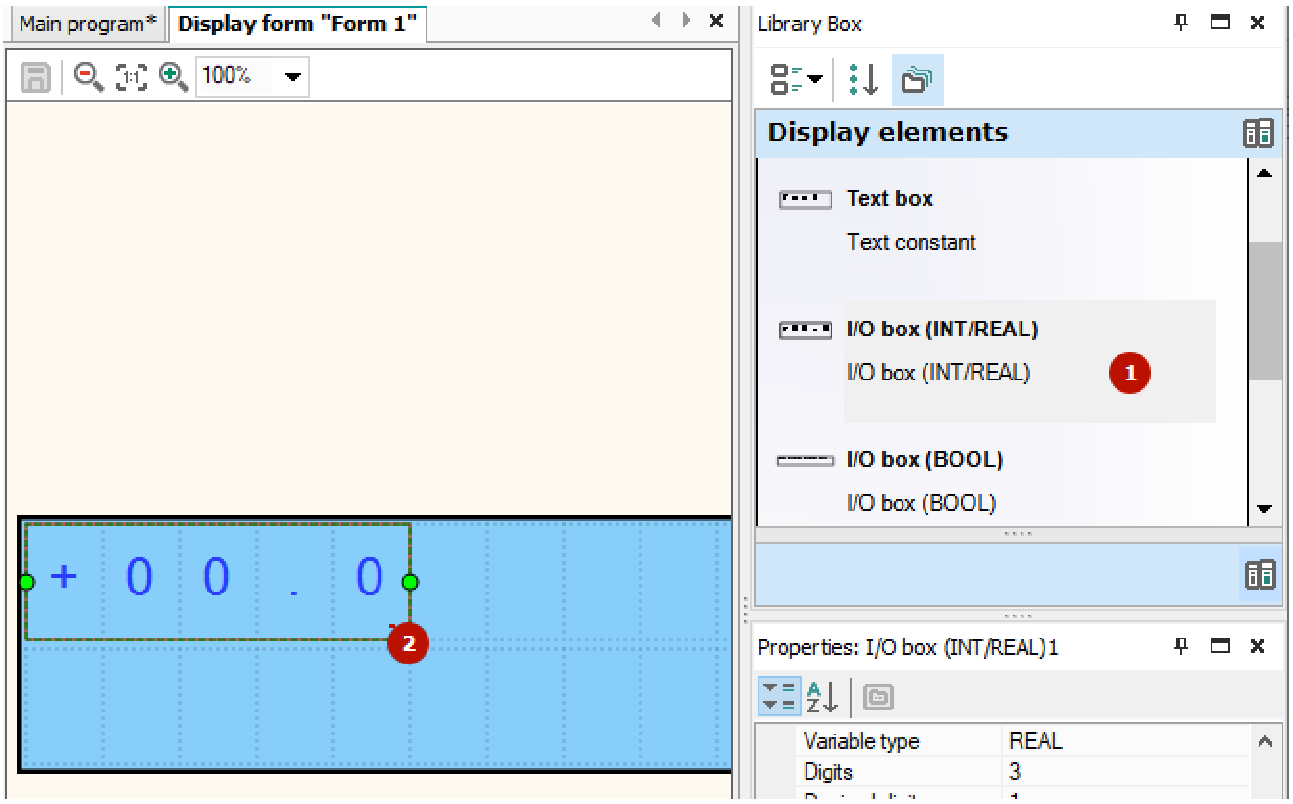


Compatible with
SMI200

Double-click on the *Display elements* folder in the *Library box* window at the top-right corner of the *akYtec ALP* window:



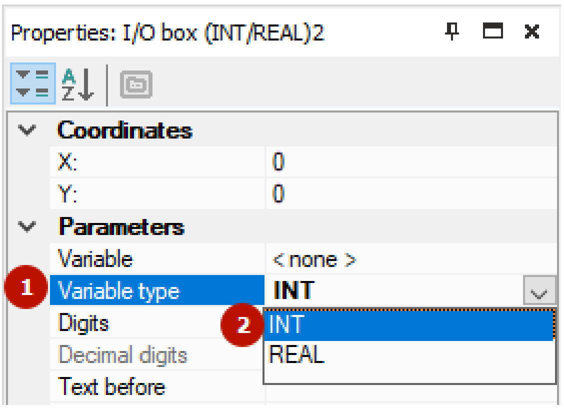
Drag and drop the *I/O box (INT/REAL)* item from the resulting list into the display form:



Variable type	REAL
Digits	3

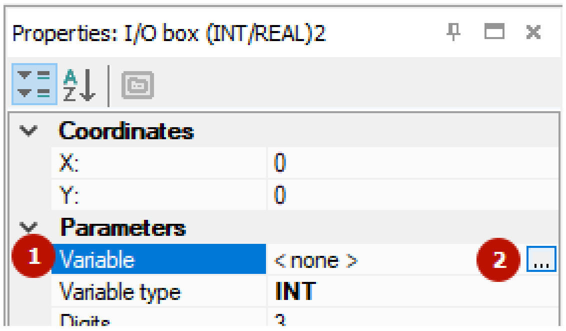
Now you can link this box with any variable in the project, say the *SETPOINT* network variable which we created in STEP 1.

In the properties section of the *I/O box (INT/REAL)* display element added, left-click on the *Variable type* parameter, left-click on the resulting ⌄ icon, and select *INT* on the resulting dropdown:


Properties: I/O box (INT/REAL)2

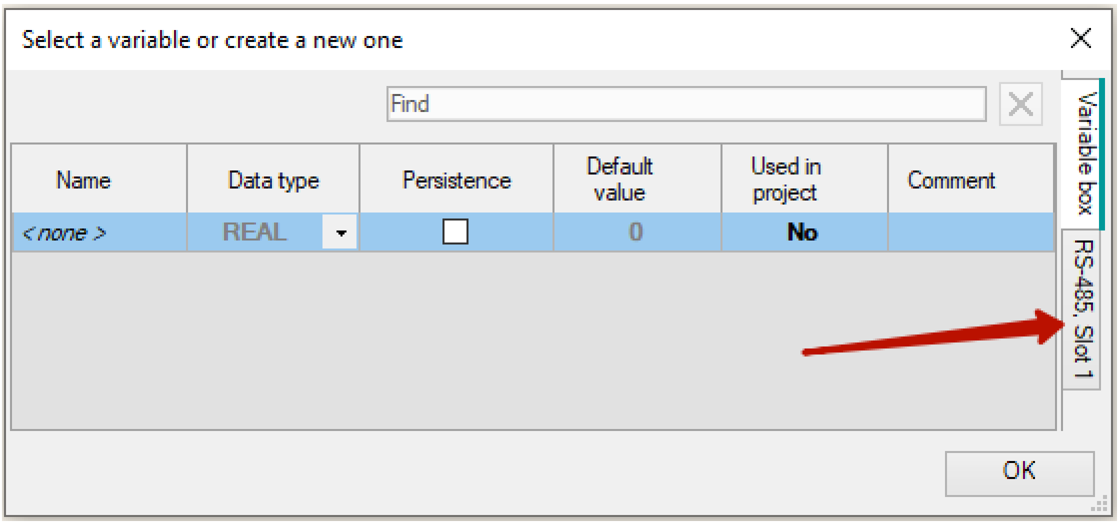
Coordinates	
X:	0
Y:	0
Parameters	
Variable	< none >
Variable type	**INT**
Digits	INT
Decimal digits	REAL
Text before	

In the very same properties section, left-click on the *Variable* parameter and then on the resulting … icon:


Properties: I/O box (INT/REAL)2

Coordinates	
X:	0
Y:	0
Parameters	
Variable	< none >
Variable type	**INT**
Digits	3

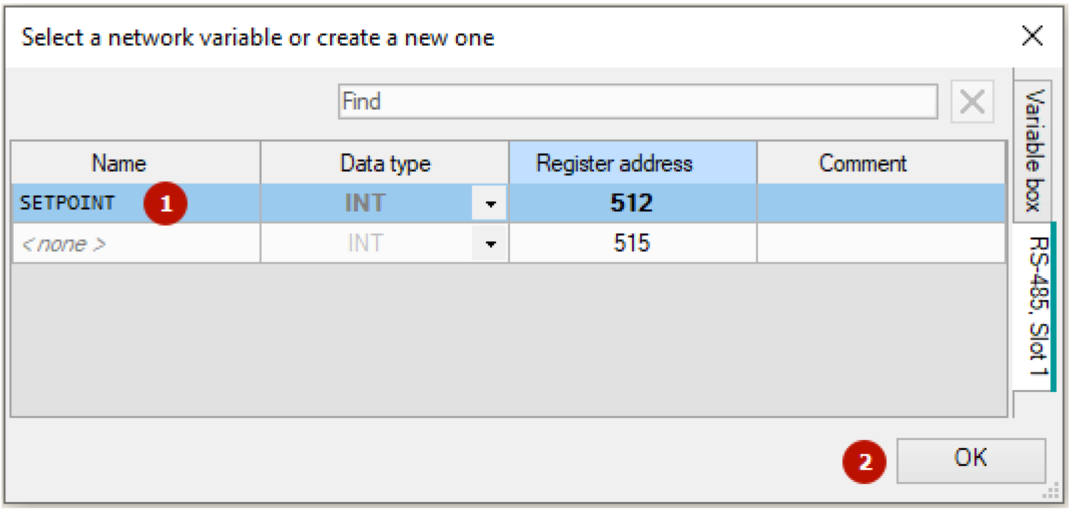
Left-click on the *RS-485, Slot 1* tab in the resulting *Select a variable or create a new one* dialog:


Select a variable or create a new one

Find

Name	Data type	Persistence	Default value	Used in project	Comment
< none >	REAL	☐	0	No	

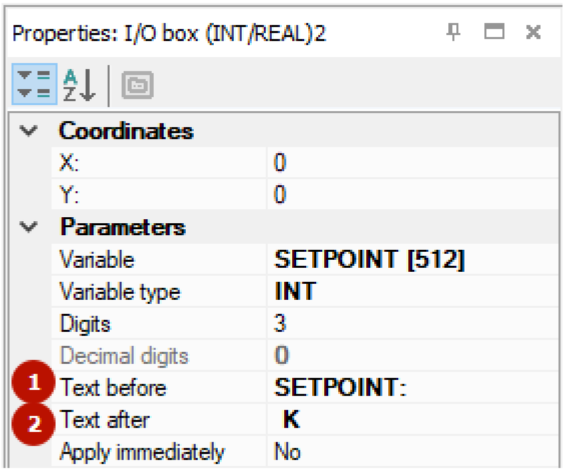
Select the *SETPOINT* variable and confirm your choice by left-clicking on the *OK* button:



Select a network variable or create a new one

Name	Data type	Register address	Comment
SETPOINT	INT	512	
< none >	INT	515	

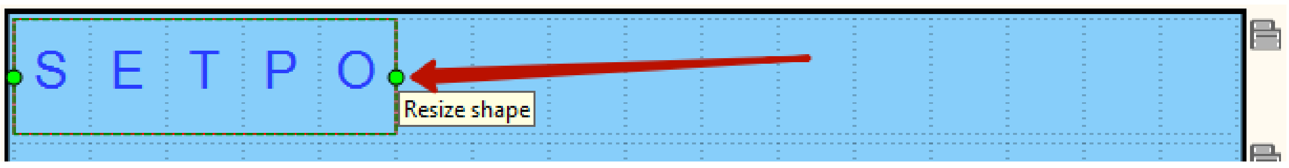
Using the *Text before* and *Text after* parameters, you can also add a bit of text for better readability and the operator's convenience. For example:

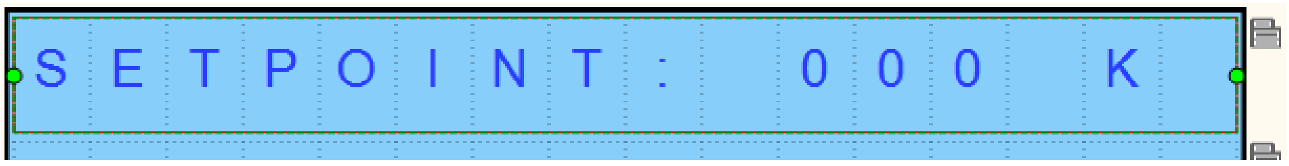


Properties: I/O box (INT/REAL)2

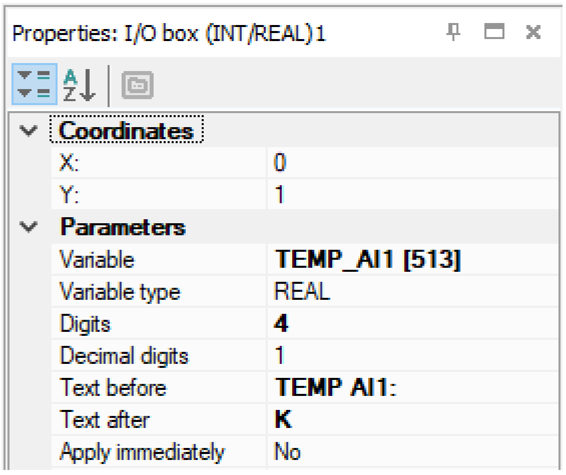
Coordinates	
X:	0
Y:	0
Parameters	
Variable	SETPOINT [512]
Variable type	INT
Digits	3
Decimal digits	0
Text before	SETPOINT:
Text after	K
Apply immediately	No

To make the whole row visible, left-click on a green disk and drag it forwards:





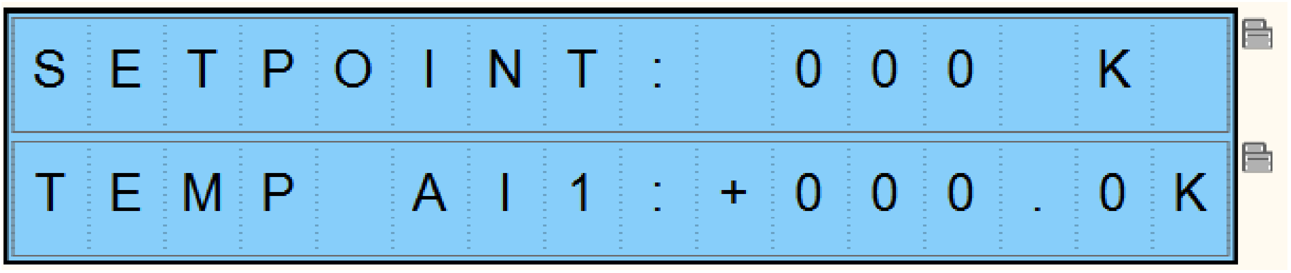
In a very similar manner, you can link the *TEMP_AI1* variable with another *I/O box (INT/REAL)* to make this variable also visible on SMI200’s LCD:



Properties: I/O box (INT/REAL)1

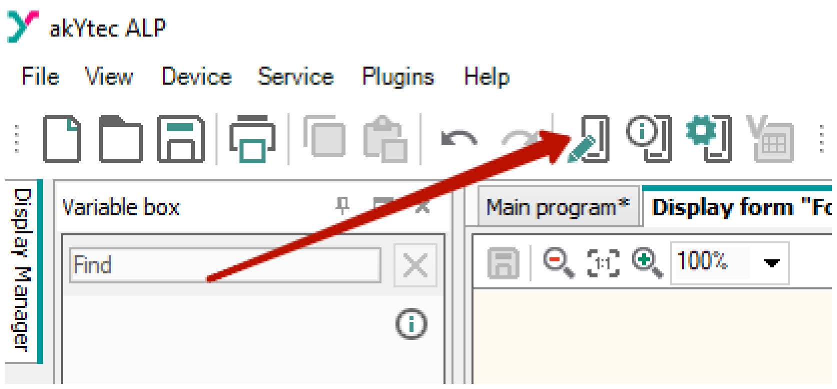
Coordinates	
X:	0
Y:	1
Parameters	
Variable	**TEMP_AI1 [513]**
Variable type	REAL
Digits	**4**
Decimal digits	1
Text before	**TEMP AI1:**
Text after	**K**
Apply immediately	No

Note that we changed the *Digits* parameter from 3 to 4, so that even real numbers over 100 would be visible:



```
SETPOINT: 000 K
TEMP AI1:+000.0K
```

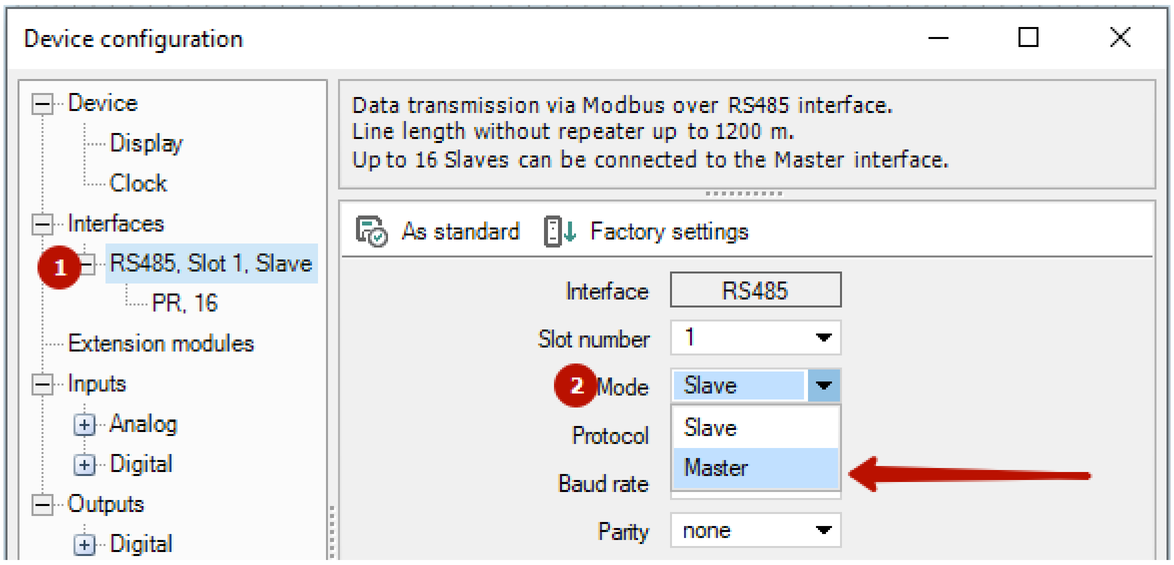
Now that all the essential and some extra configuration steps of SMI200 to operate as a Slave device have been finished, you should transfer these settings to the device memory by left-clicking on the icon in the toolbar:



4.2 Configuration of PR200 to operate as Master

STEP 1 Master identification

Having connected your PR200 to the PC and established a successful communication between them (see Section 3), open the *Device configuration* dialog by clicking on the *Device configuration* icon in the toolbar and ensure that the (1) device interface is in the Master mode:

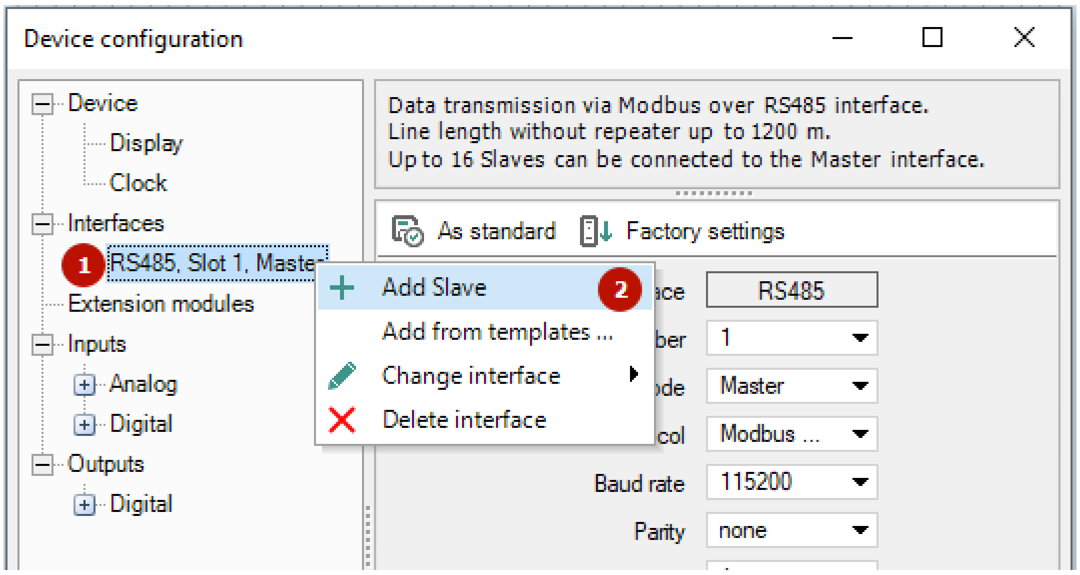


Data transmission via Modbus over RS485 interface.
Line length without repeater up to 1200 m.
Up to 16 Slaves can be connected to the Master interface.

Interface	RS485
Slot number	1
Mode	Slave
Protocol	
Baud rate	
Parity	none

STEP 2 Identification of Slave devices

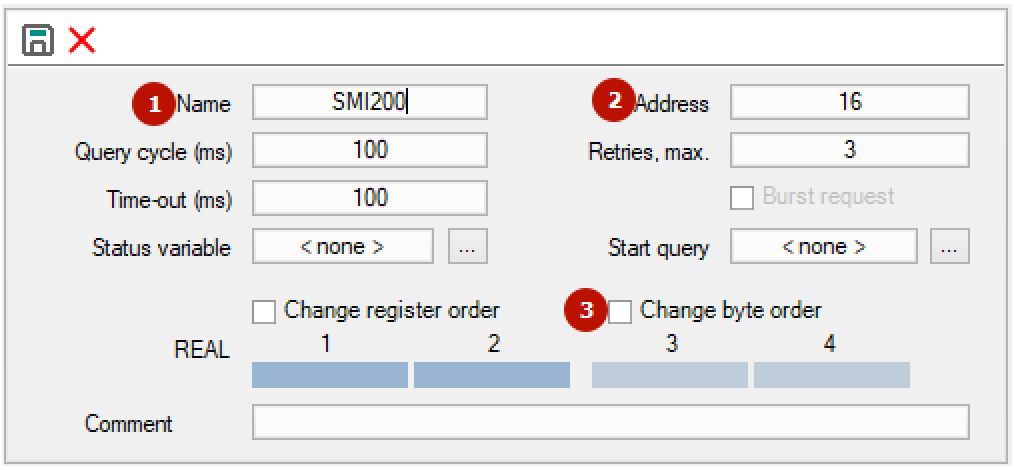
Add your SMI200 as a Slave device by right-clicking on the *RS485, Slot 1, Master* item and left-clicking on the *Add Slave* item on the resulting option list:



Data transmission via Modbus over RS485 interface.
Line length without repeater up to 1200 m.
Up to 16 Slaves can be connected to the Master interface.

...ce	RS485
...ber	1
...de	Master
...col	Modbus ...
Baud rate	115200
Parity	none

Now you can fill in all the essential information which will assist the PR200 (Master) to identify the SMI200 (Slave):



1 Name	SMI200	2 Address	16
Query cycle (ms)	100	Retries, max.	3
Time-out (ms)	100		☐ Burst request
Status variable	< none >	Start query	< none >

☐ Change register order 3 ☐ Change byte order

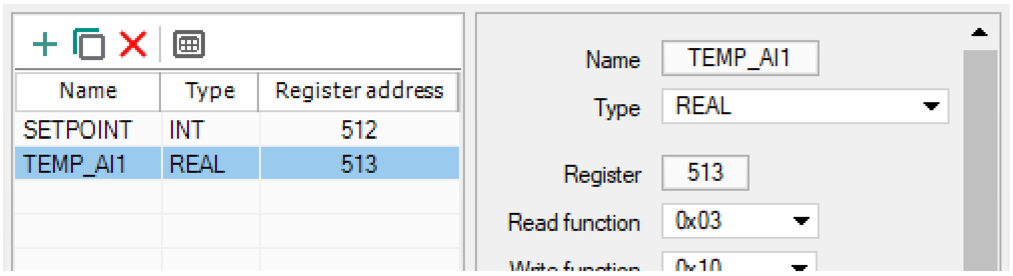
REAL 1 2 3 4

Comment

(1) ***Name*** – You can choose any reasonable name to put here. In this case, *SMI200* is reasonable enough.
(2) ***Address*** – This number must match the address of a corresponding slave device. In our case it is *16* because we gave it to our SMI200 here (marked blue, Section 4.1.1, Step 1).
(3) Since we are going to communicate values not only through INT variables but also through REAL ones, you must uncheck the ***Change byte order*** checkbox. The thing is that SMI200's firmware is designed to process network variables of type REAL using big endian byte order.

STEP 3 Declaration of network variables

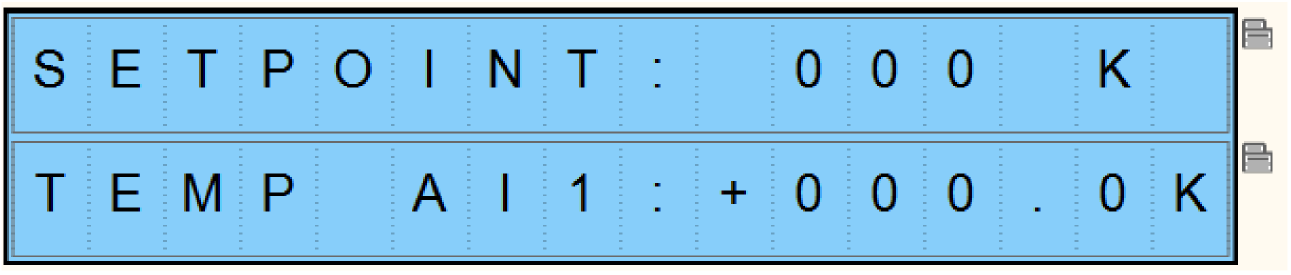
Just like in Section 4.1, Step 2, you need to declare some network variables and link them with those variables on the Slave's side. The main linking chain is the *Register* parameter in this case. Two variables are linked if they have the same register address and the same type, yet their names could differ. You should have something similar to this:



Name	Type	Register address
SETPOINT	INT	512
TEMP_AI1	REAL	513

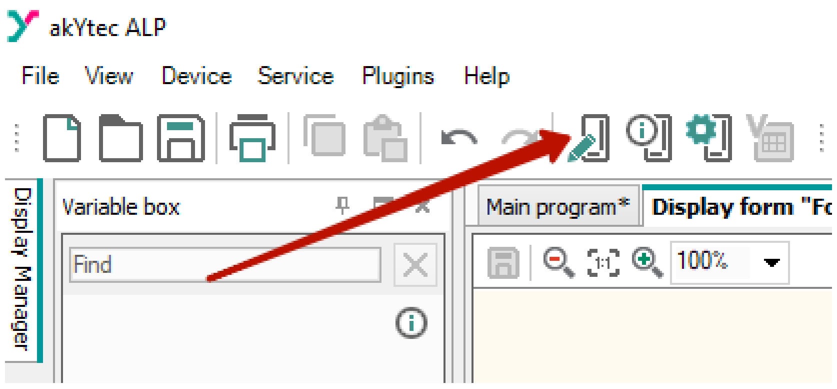
Name	TEMP_AI1
Type	REAL
Register	513
Read function	0x03
Write function	0x10

At this point, the PR200 has already been configured enough to operate as the Modbus Master device. However, to allow you to monitor the above-declared variables and assign them some values with the function keys on the device front, let us create a simple user interface on PR200's LCD first. You can refer to STEP 3 in Section 4.1 since the procedure is absolutely similar. Upon finishing, you will have something similar to this:



```
SETPOINT:  000  K
TEMP  AI1:+000.0K
```

Now that all the essential and some extra configuration steps of PR200 to operate as the Master device have been finished, you should transfer these settings to the device memory by left-clicking on the icon in the toolbar:



Now that you have configured both devices, it is time to physically connect them with wires. After you have finished wiring according to *Figure 1* in *Section 5* and supply power to the circuit, you will be able to enter some values using the function keys either of SMI200 or PR200 and see them on the corresponding device display.

5 Wiring

CAUTION Pay attention to the supply voltage of your PR200. Do not connect a PR200-**24**.X.X to a 230 V AC power supply.

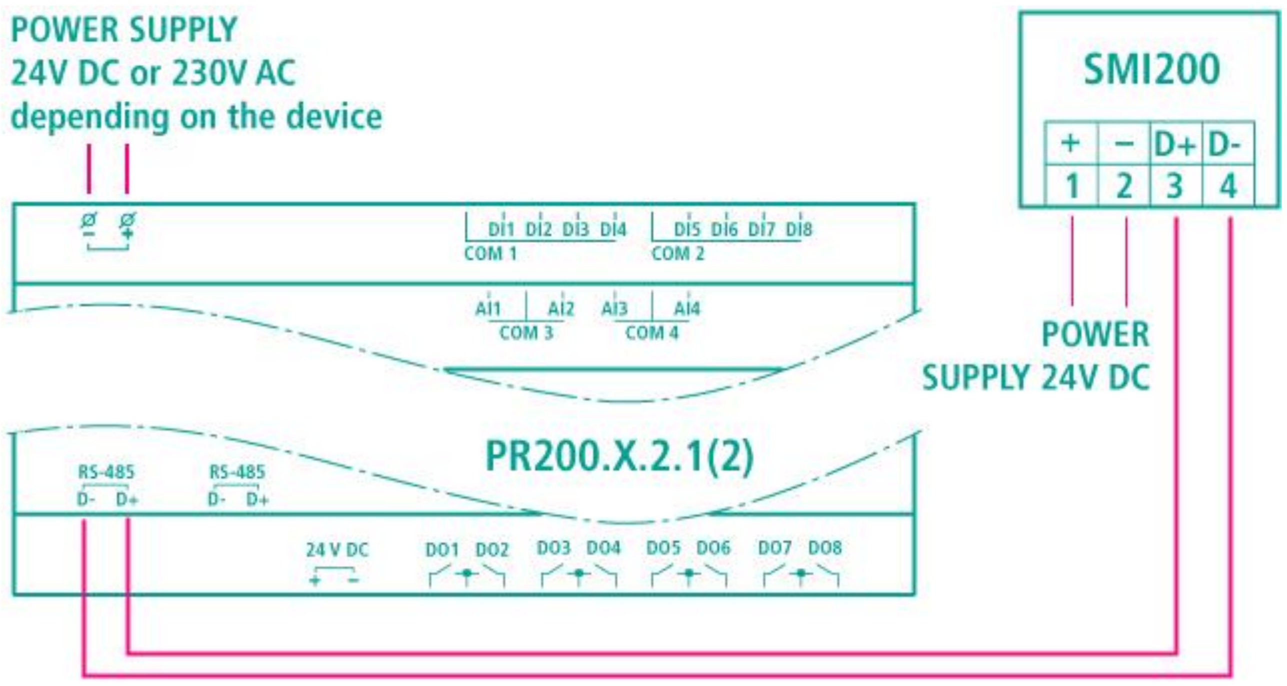


Figure 1

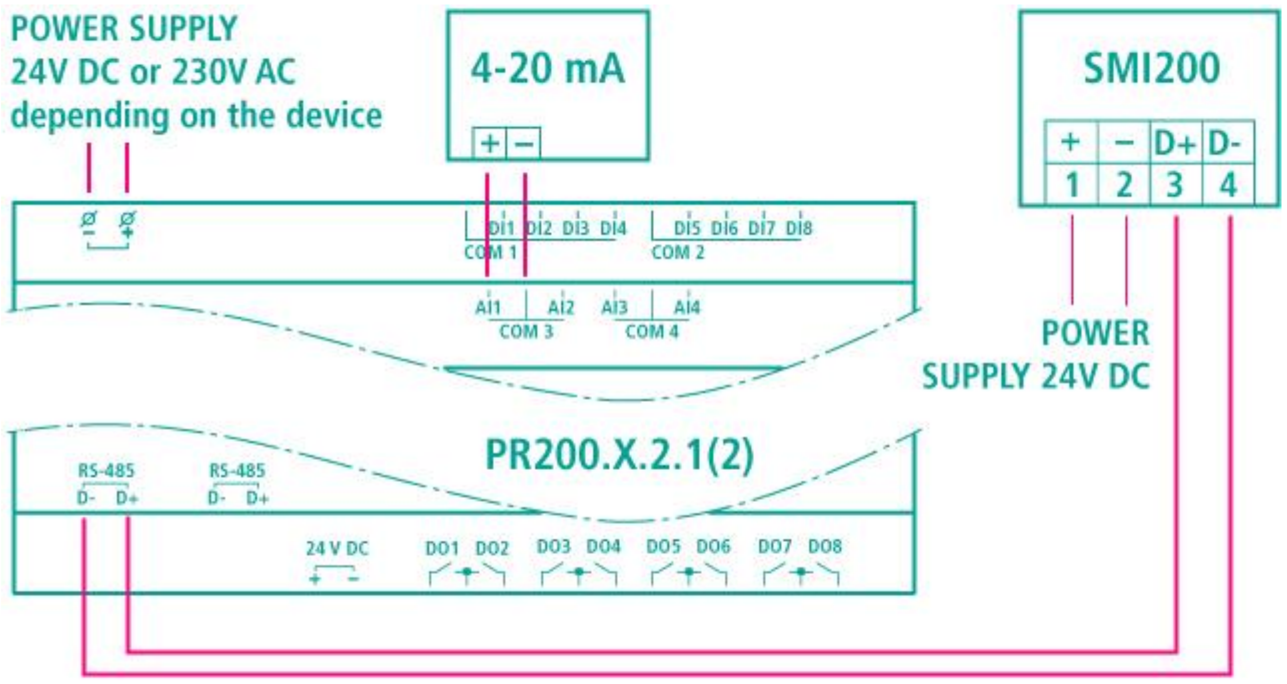


Figure 2

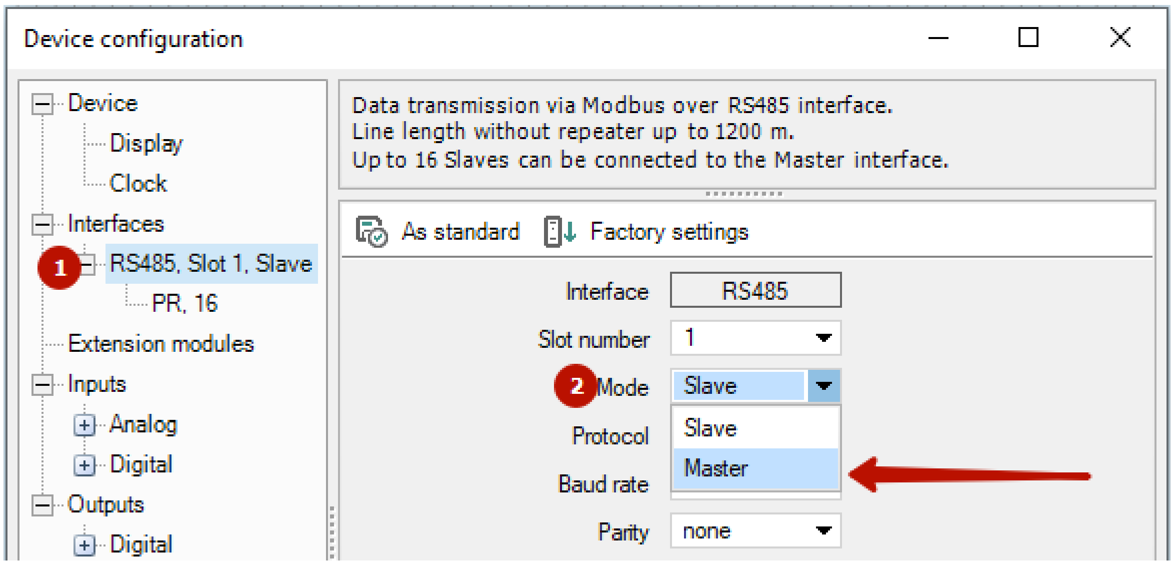
Bonus: Direct access to Modbus registers of a Slave device with SMI200 operating as Master

Again, there is a control cabinet enclosure with a PR200 snapped on a DIN rail inside. You cannot access the PR200's LCD and its function buttons unless you open the enclosure door. Nonetheless, you had the forethought to mount a SMI200 at this door, wire it with the PR200, and establish proper communication between them.

Assume that now not only want you to check values of some network variables using this SMI200 but also you want to read an analog input of the PR200. Since the present project in that PR200 is already very complex, the situation is worsening – there has left no more memory room in the device to declare a network variable, even a single one. The only way to overcome this obstacle is to access to that analog input directly through its Modbus address. This is exactly what this section is about.

STEP 1 Configuration of the Master device (SMI200)

Having connected your SMI200 to the PC and established a successful communication between them (see Section 3), open the *Device configuration* dialog by clicking on the *Device configuration* icon in the toolbar and ensure that the (1) *RS485, Slot 1* interface is in the Master mode:

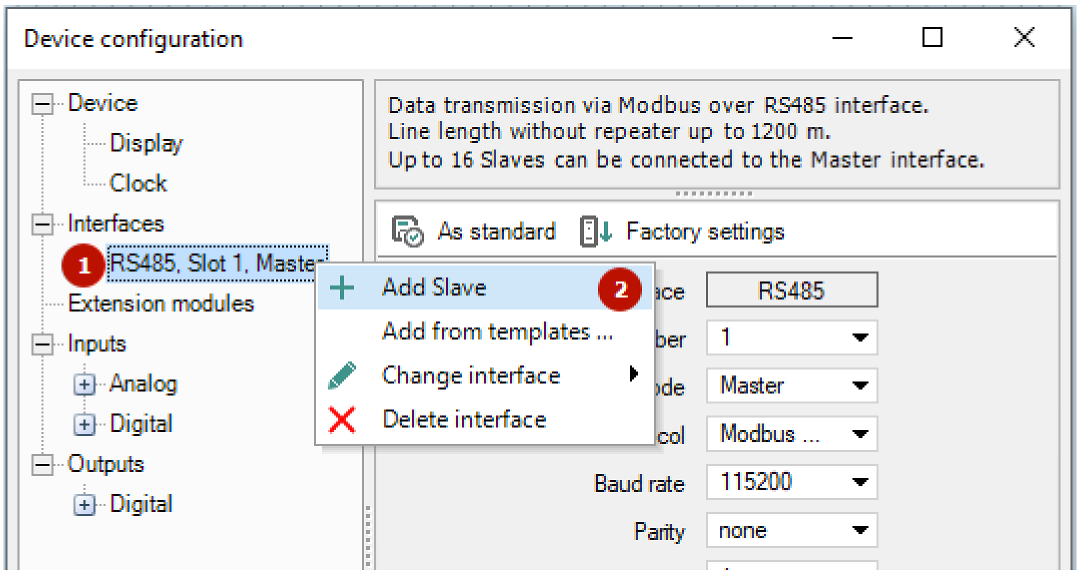


Data transmission via Modbus over RS485 interface.
Line length without repeater up to 1200 m.
Up to 16 Slaves can be connected to the Master interface.

Interface: RS485
Slot number: 1
Mode: Slave
Protocol
Baud rate
Parity: none

STEP 2 Identification of Slave devices

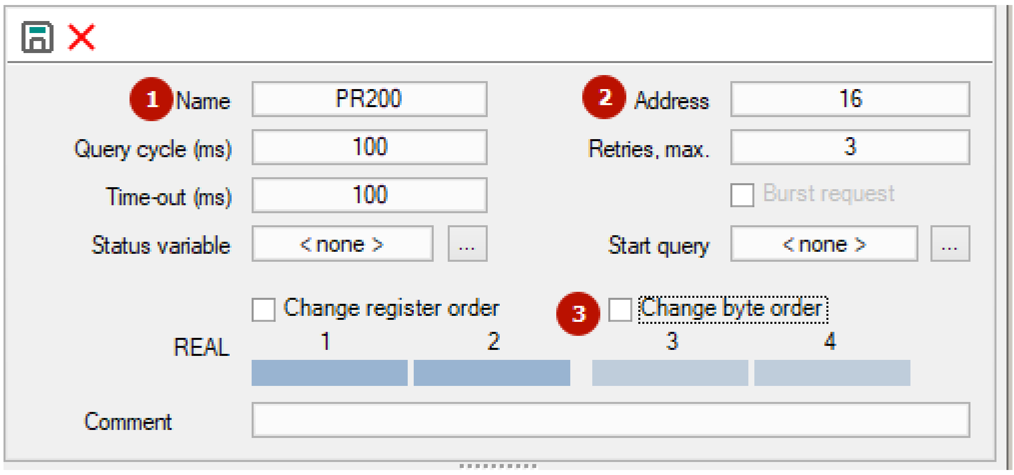
Add your PR200 as a Slave device by right-clicking on the *RS485, Slot 1, Master* item and left-clicking on the *Add Slave* item on the resulting option list:



Data transmission via Modbus over RS485 interface.
Line length without repeater up to 1200 m.
Up to 16 Slaves can be connected to the Master interface.

...ace	RS485
...ber	1
...ode	Master
...col	Modbus ...
Baud rate	115200
Parity	none

Now you can fill in all the essential information which will assist the SMI200 (Master) to identify the PR200 (Slave):



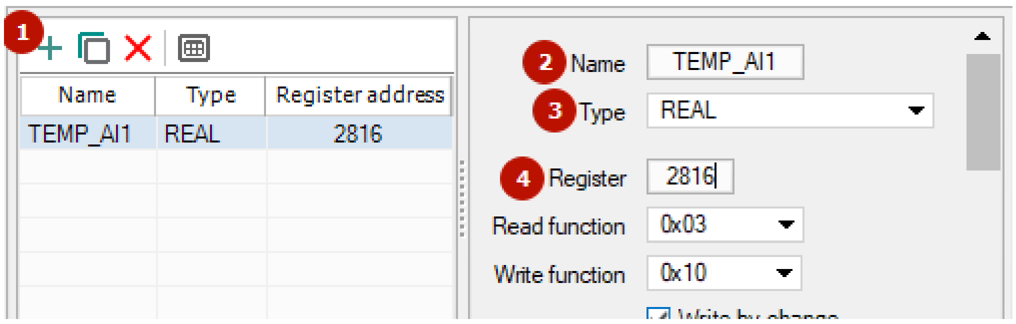
Name	PR200	Address	16
Query cycle (ms)	100	Retries, max.	3
Time-out (ms)	100		☐ Burst request
Status variable	< none >	Start query	< none >
	☐ Change register order		☐ Change byte order
Comment			

(1) ***Name*** – You can choose any reasonable name to put here. In this case, *PR200* is reasonable enough.
(2) ***Address*** – This number must match the address of a corresponding slave device. In our case it is *16* because PR200 has it by default.
(3) Since we are going to communicate values using a variable of type REAL, you must uncheck the ***Change byte order*** checkbox. The thing is that PR200's firmware is designed to process network variables of type REAL using big endian byte order.

Compatible with
SMI200

STEP 3 Declaration of network variables

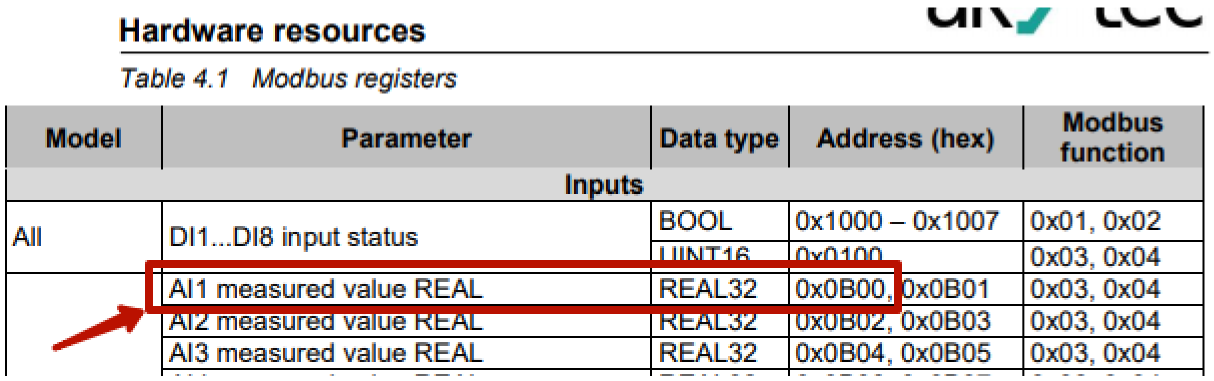
You need a variable which is to receive and contain values from an analog input of the Slave device (PR200), so (1) declare a network variable of (3) type REAL with (2) an arbitrary name:



Name	Type	Register address
TEMP_AI1	REAL	2816

Name: TEMP_AI1
Type: REAL
Register: 2816
Read function: 0x03
Write function: 0x10

(3) ***Type*** – We use REAL in this section to provide you with practical skills and abilities needed to cope with this type of data which is usually more troublesome than the type INT. In your real projects, you are up to select from BOOL, INT, and REAL.
(4) ***Register*** – Here you need to fill in the register address of the particular Slave's input which you want to poll. For example, if you want to read the measured value at analog input 1 of the PR200 using data type REAL, register address 0x0B00 must be used according to User guide, Table 4.1, page 17:



Hardware resources

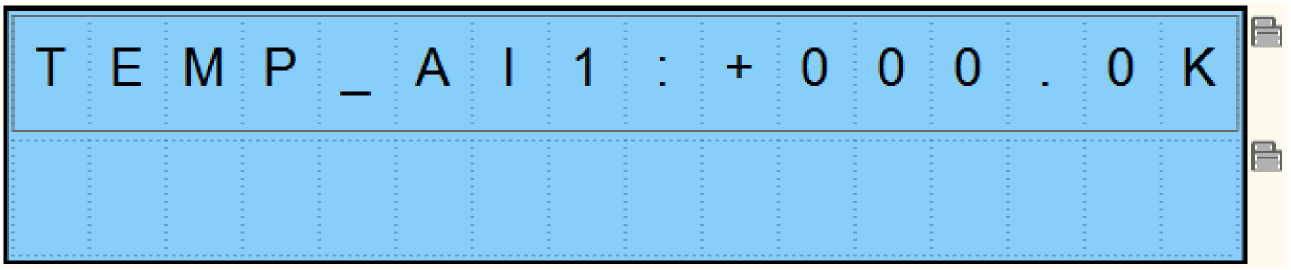
Table 4.1 Modbus registers

Model	Parameter	Data type	Address (hex)	Modbus function
Inputs				
All	DI1...DI8 input status	BOOL	0x1000 – 0x1007	0x01, 0x02
		UINT16	0x0100	0x03, 0x04
	AI1 measured value REAL	REAL32	0x0B00, 0x0B01	0x03, 0x04
	AI2 measured value REAL	REAL32	0x0B02, 0x0B03	0x03, 0x04
	AI3 measured value REAL	REAL32	0x0B04, 0x0B05	0x03, 0x04

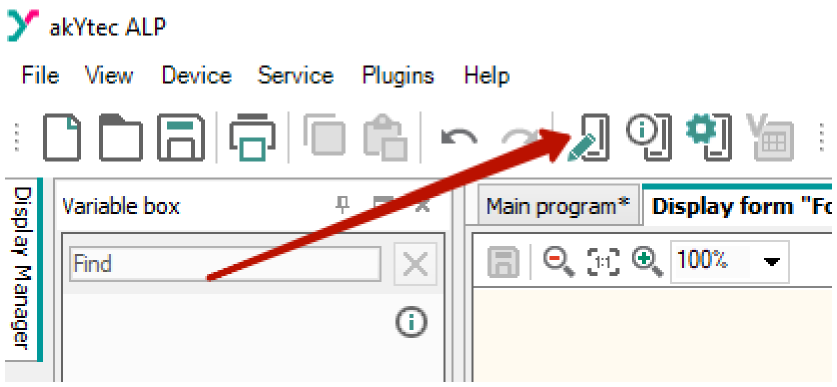
Since the ***Register*** parameter in this dialog must be entered using the decimal numeral system, you need to use *2816*.

NOTE	When adding network variables in the *Device configuration* dialog in akYtec ALP, you must define the Register parameter <u>using the decimal numeral system</u>

At this point, the SMI200 has already been configured enough to operate as the Modbus Master device. However, to allow you to monitor the above-declared variable and assign it some values with the function keys on the device front, let us create a simple user interface on SMI200's LCD first. You can refer to STEP 3 in Section 4.1 since the procedure is absolutely similar. Upon finishing, you will have something similar to this:

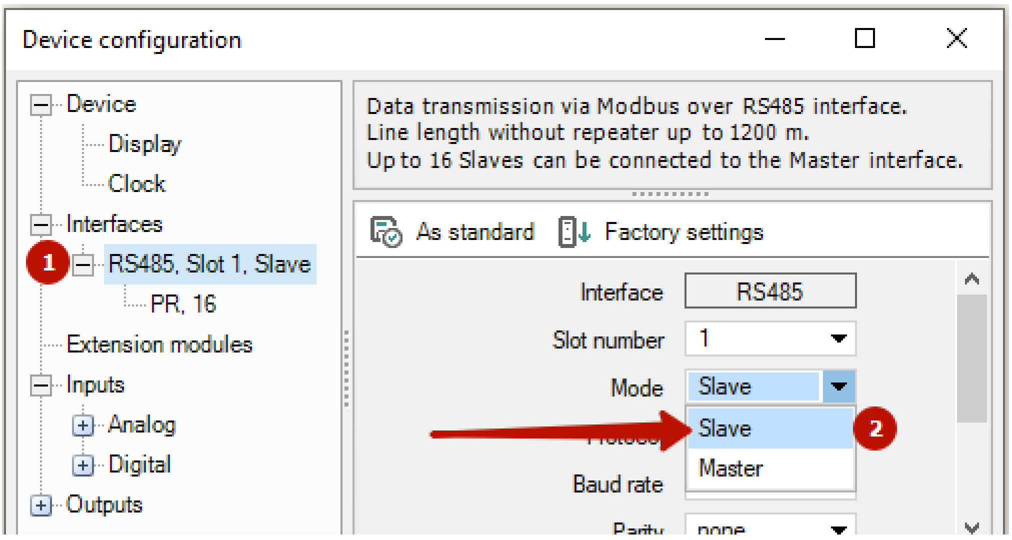


Now that all the essential and some extra configuration steps of SMI200 to operate as the Master device have been finished, you should transfer these settings to the device memory by left-clicking on the icon in the toolbar:



STEP 4 Configuration of a Slave device (PR200)

Having connected your PR200 to the PC and established a successful communication between them (see Section 3), open the *Device configuration* dialog by clicking on the *Device configuration* icon in the toolbar and ensure that the (1) device interface is in the Slave mode:

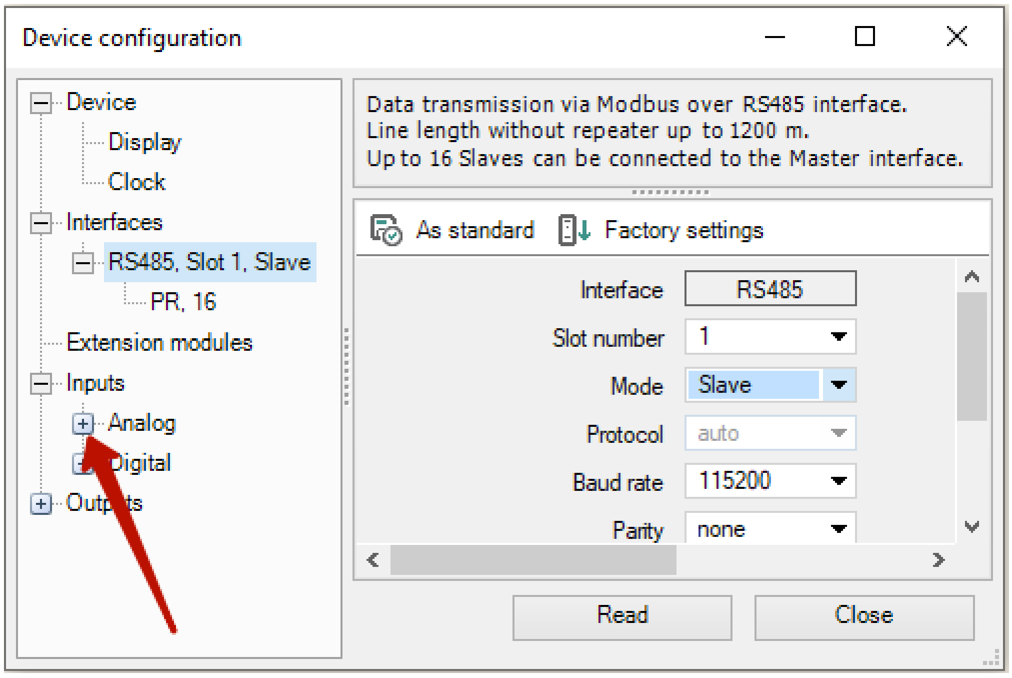


Data transmission via Modbus over RS485 interface.
Line length without repeater up to 1200 m.
Up to 16 Slaves can be connected to the Master interface.

Interface	RS485
Slot number	1
Mode	Slave
Protocol	Slave / Master
Baud rate	
Parity	none

Otherwise, select **Slave** on the (2) *Mode* dropdown.

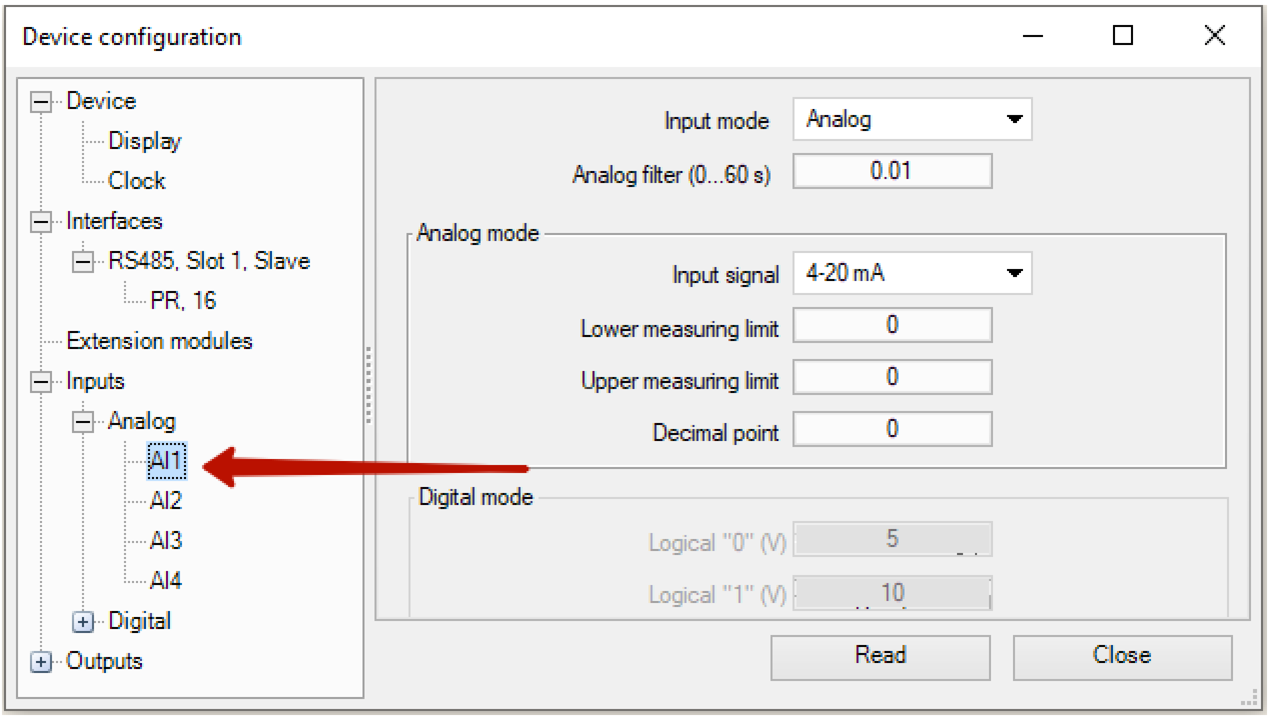
Now you need to configure analog input 1 of your PR200. In the tree to the left of the *Device configuration* window, left-click on the icon in front of *Analog* under the *Inputs* node:



Data transmission via Modbus over RS485 interface.
Line length without repeater up to 1200 m.
Up to 16 Slaves can be connected to the Master interface.

Interface	RS485
Slot number	1
Mode	Slave
Protocol	auto
Baud rate	115200
Parity	none

Left-click on the *AI* item in the resulting dropdown:


Device configuration

- Device
  - Display
  - Clock
- Interfaces
  - RS485, Slot 1, Slave
    - PR, 16
- Extension modules
- Inputs
  - Analog
    - AI1
    - AI2
    - AI3
    - AI4
  - Digital
- Outputs

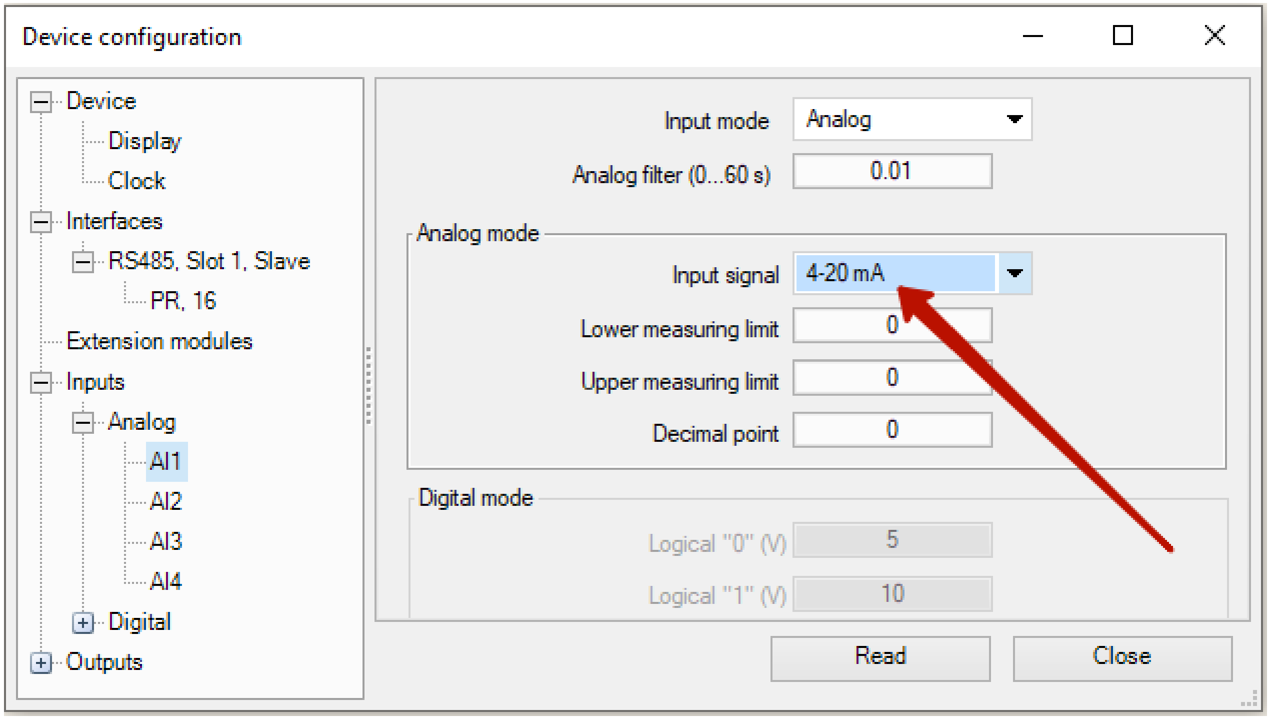
Input mode: Analog
Analog filter (0...60 s): 0.01

Analog mode
Input signal: 4-20 mA
Lower measuring limit: 0
Upper measuring limit: 0
Decimal point: 0

Digital mode
Logical "0" (V): 5
Logical "1" (V): 10

Since we suggest using anything that can generate a signal of 4-20 mA in Section 1 of this sample project, we are going to stick to this type of signal. In your real projects, you are up to select from 4-20 mA, 0-10 V, and 0-4000 ohm.

Ensure that the *Input signal* parameter is 4-20 mA:


Device configuration

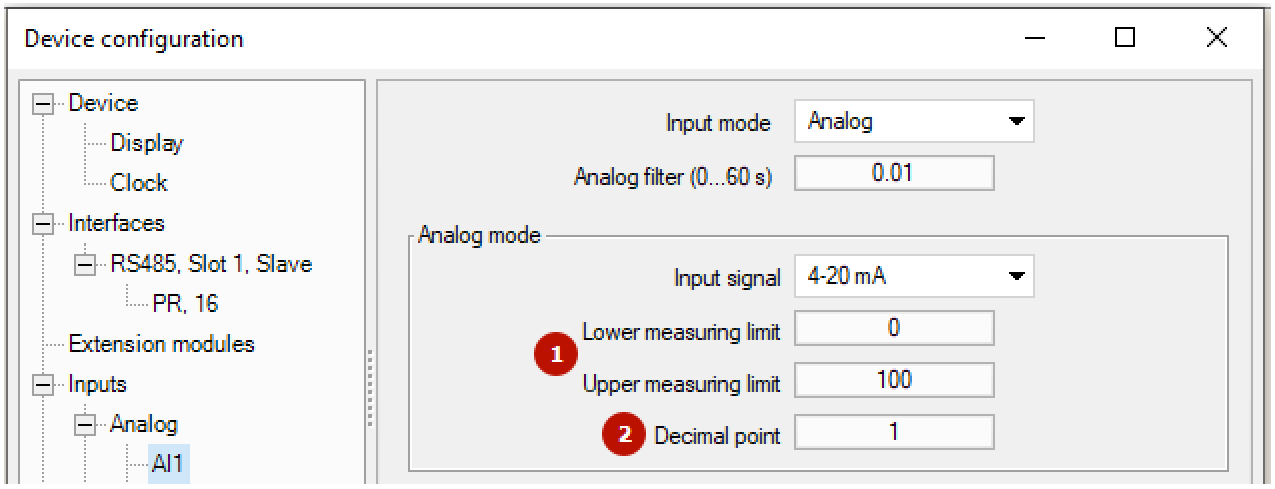
- Device
  - Display
  - Clock
- Interfaces
  - RS485, Slot 1, Slave
    - PR, 16
- Extension modules
- Inputs
  - Analog
    - AI1
    - AI2
    - AI3
    - AI4
  - Digital
- Outputs

Input mode: Analog
Analog filter (0...60 s): 0.01

Analog mode
Input signal: 4-20 mA
Lower measuring limit: 0
Upper measuring limit: 0
Decimal point: 0

Digital mode
Logical "0" (V): 5
Logical "1" (V): 10

Now you need to scale the measuring limits using (1) the parameters *Lower measuring limit* and *Upper measuring limit*. Fox example, it could be 0-100:


Device configuration

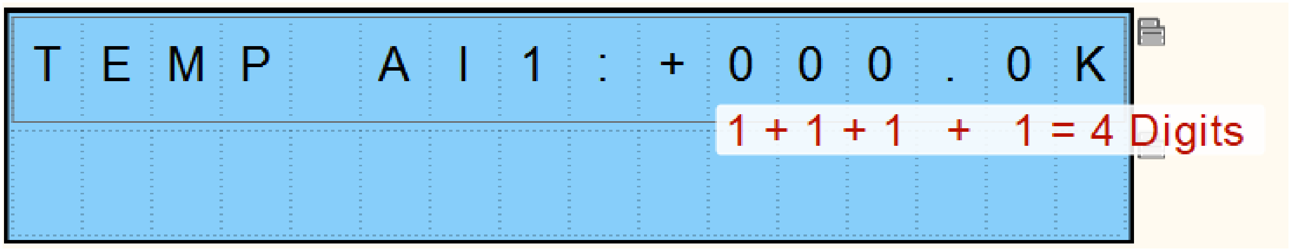
- Device
  - Display
  - Clock
- Interfaces
  - RS485, Slot 1, Slave
    - PR, 16
- Extension modules
- Inputs
  - Analog
    - AI1

Input mode	Analog
Analog filter (0...60 s)	0.01

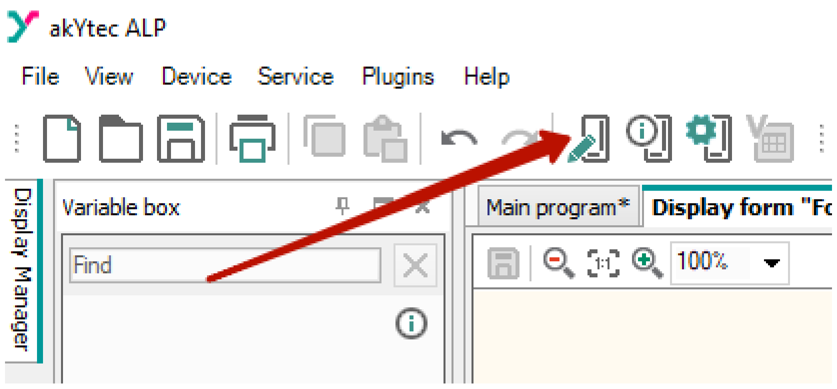
Analog mode

Input signal	4-20 mA
1 Lower measuring limit	0
Upper measuring limit	100
2 Decimal point	1

(2) ***Decimal point*** – this parameter defines how many digits are to be displayed to the right of the decimal point. Having defined a total of 4 digits to display the value of the *TEMP_AI* variable while configuring the user interface of the SMI200's LCD and having the upper measuring limit of 100, now we better not define the Decimal point parameter as a number more than *1* to have the correct display of values:



Now that all the essential configuration steps of the PR200 to operate as a Slave device have been finished, you should transfer these settings to the device memory by left-clicking on the icon in the toolbar:



Now that you have configured both devices, it is time to physically connect them with wires. After you have finished wiring according to *Figure 2* in *Section 5* and supply power to the circuit, you will see, depending on the current magnitude at the PR200's analog input 1, a decimal fraction in the range from 0.X to 100.X on the SMI200's LCD.