M171O Modbus Communication

TM171ODM22R Master & Slave configuration

Aidin Aliyarzadeh
Technical Trainer
Marktheidenfeld/Germany, June 2016

Schneider Electric

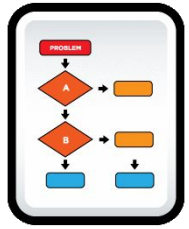
Agenda

- Final Destination/Function Description
- Hardware
- Slave Configuration
- Master Configuration
- 1st Method: Generic Modbus RTU
- 2nd Method: Target Blocks
- Modbus Monitoring & Debugging
- Appendix 1, Function Codes

Final Destination

Function Description

Exercise Goal



Function Description

1. Read Slave's NTC probe value by Master controller via Modbus Serial Line.
2. Write to Slave's EEPROM Parameters by Master controller via Modbus Serial Line.
3. Enabling Modbus Master by related Function Block.
4. Enable/Disable Enumerators creation.
5. Menu creation to access port activation/deactivation from front face of product.
6. Modbus protocol properties configuration on both parties.
7. Continuous Modbus Communication watching/monitoring by dedicated vector.

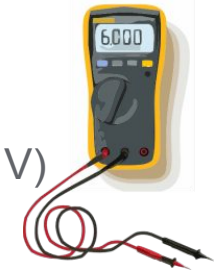
Note: It is **not** needed to configure Modbus Read/Write function codes, protocol… in Connection Tool.

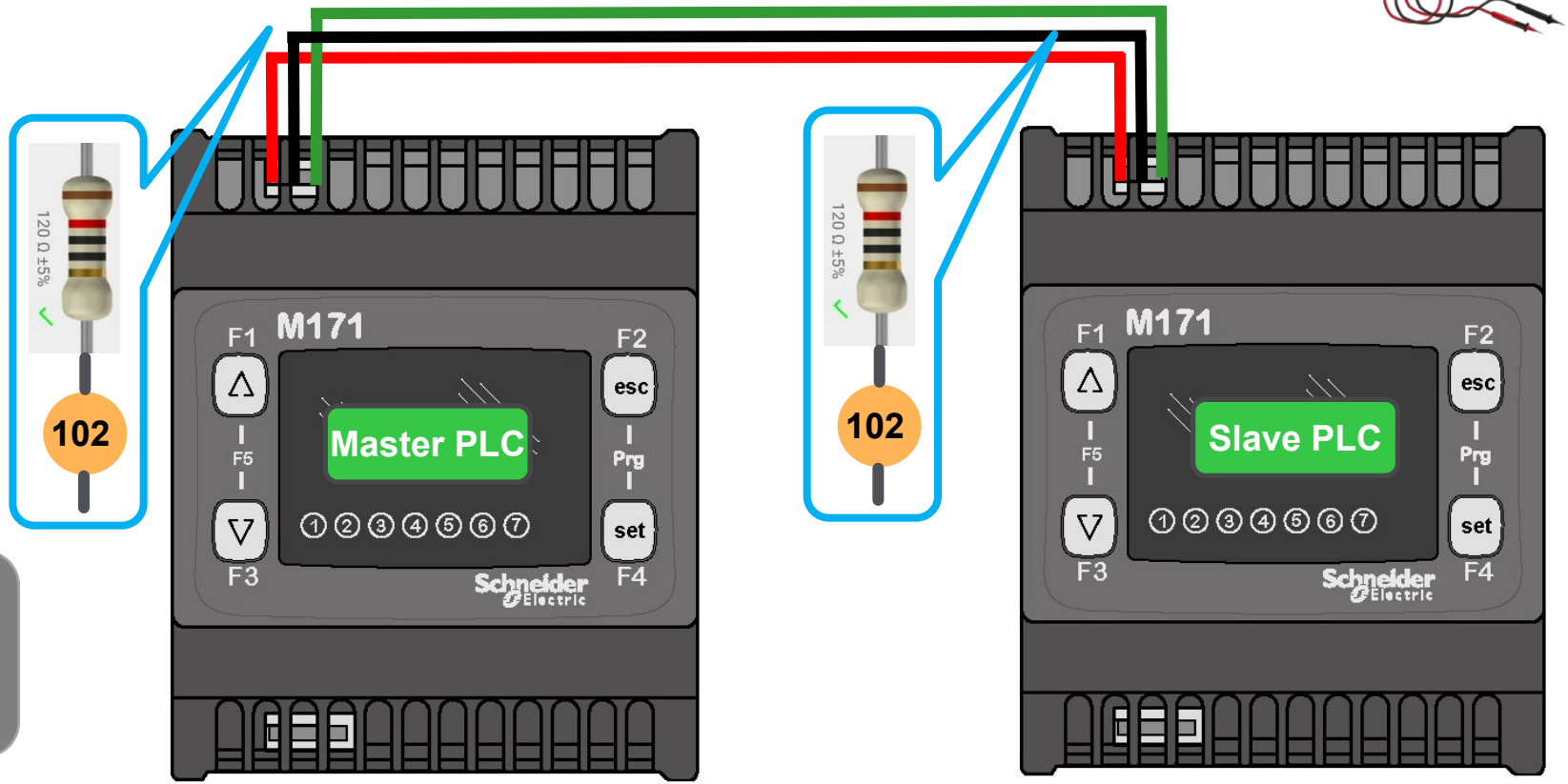
M171O Controller Hardware

TM171ODM22R Modbus wiring and termination

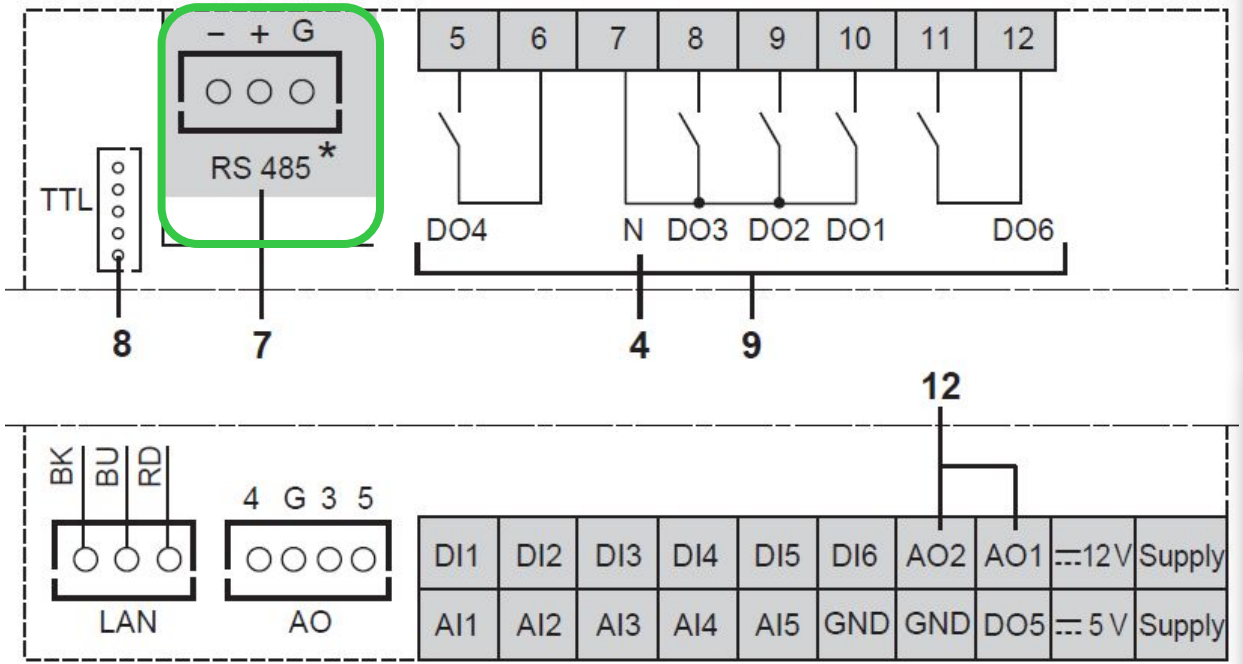
M171 Optimized controller Hardware

TM171ODM22R Modbus Wiring & Termination (120 Ohms, 0.25 W series with 1nF, 10 V)

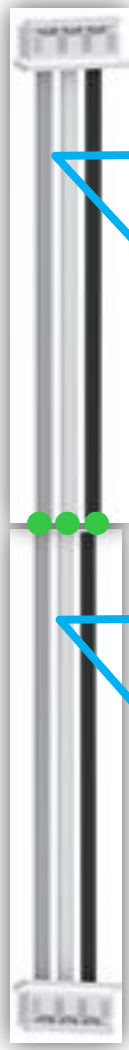




TM171ODM22R Pin-Out & Modbus wire



7 - RS485 Serial for TM171OM*****



Modbus Wire/Connection

Modbus SL connector
TM171ACB4ORS485

Cordset equipped with a 3-pin connector on one end, 1 m length

Modbus SL connector
TM171ACB4ORS485

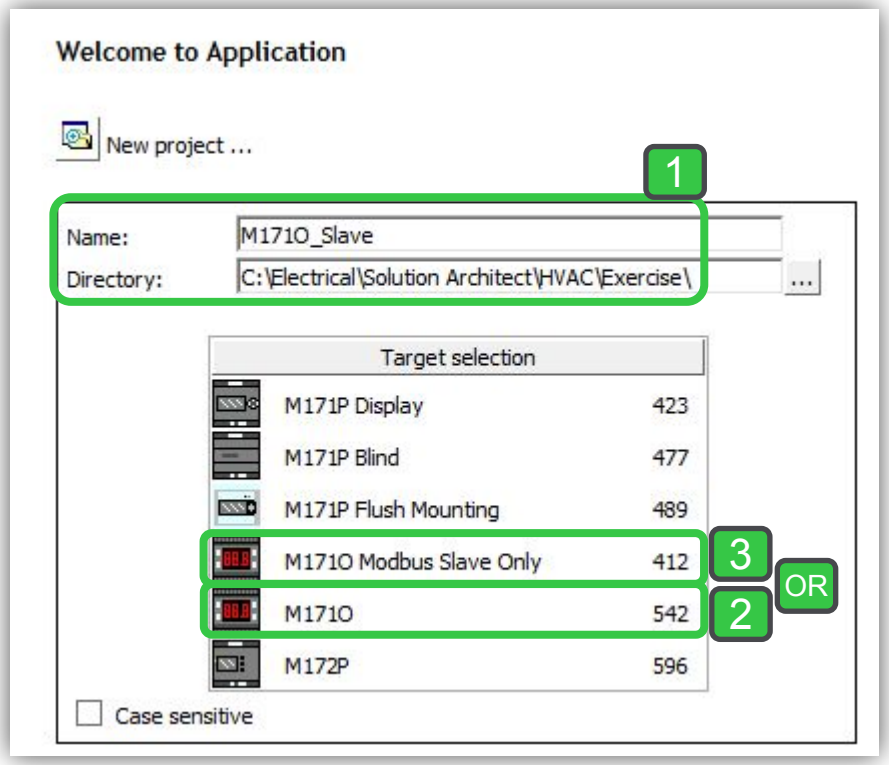
Cordset equipped with a 3-pin connector on one end, 1 m length

M171O Controller

Modbus Serial line Configuration, **Slave Side**

Creating New project

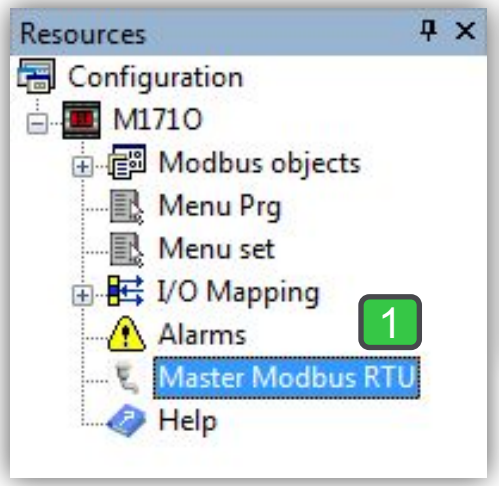
Slave Side

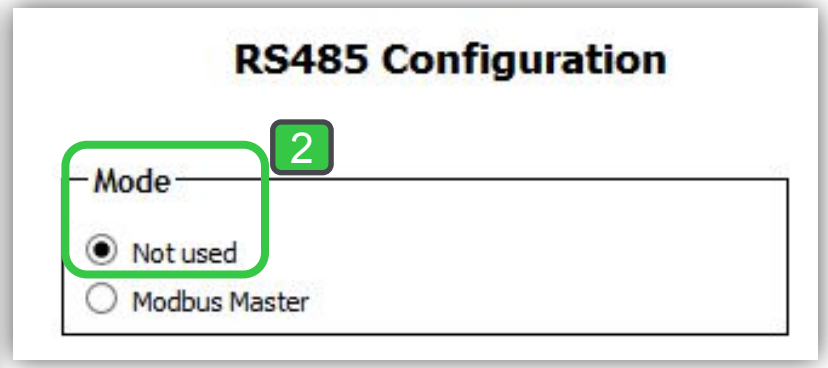


Note: As a slave both 412 or 542 mask versions controllers could be use.

Modbus RTU configuration/Slave Mode

Slave side

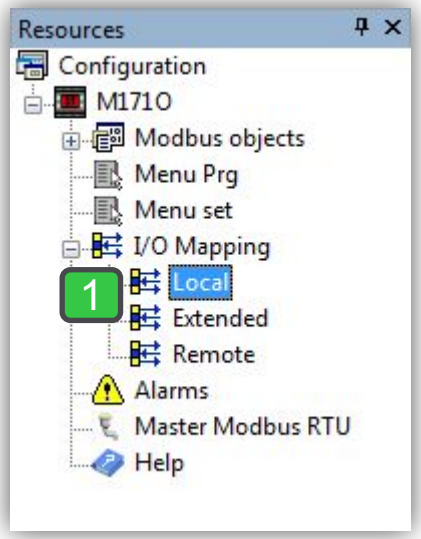




Note: Not used taken into account as Slave

Physical I/O Mapping

Slave side



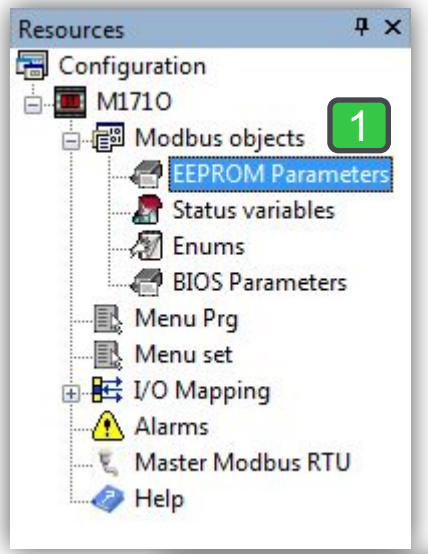
Local I/O Mapping 2

#	Name	Variable	Type	Description
1	AIL1	NTC_Probe	INT	AIL1 analogue input
2	AIL2		INT	AIL2 analogue input
3	AIL3		INT	AIL3 analogue input
4	AIL4		INT	AIL4 analogue input
5	AIL5		INT	AIL5 analogue input
6	DIL1		BOOL	DIL1 digital input
7	DIL2		BOOL	DIL2 digital input
8	DIL3		BOOL	DIL3 digital input
9	DIL4		BOOL	DIL4 digital input
10	DIL5		BOOL	DIL5 digital input
11	DIL6		BOOL	DIL6 digital input
12	DOL1	Output_Cooling	BOOL	DOL1 digital output
13	DOL2	Alarm	BOOL	DOL2 digital output
14	DOL3		BOOL	DOL3 digital output
15	DOL4		BOOL	DOL4 digital output
16	DOL5		BOOL	DOL5 digital output
17	DOL6		BOOL	DOL6 digital output
18	AOL1		INT	AOL1 analogue output
19	AOL2		INT	AOL2 analogue output
20	AOL3		INT	AOL3 analogue output
21	AOL4		INT	AOL4 analogue output
22	AOL5		INT	AOL5 analogue output
23	TCL1		INT	TCL1 analogue output

Note:

EEPROM parameters definition/list

Slave side



Note:

EEPROM Parameters

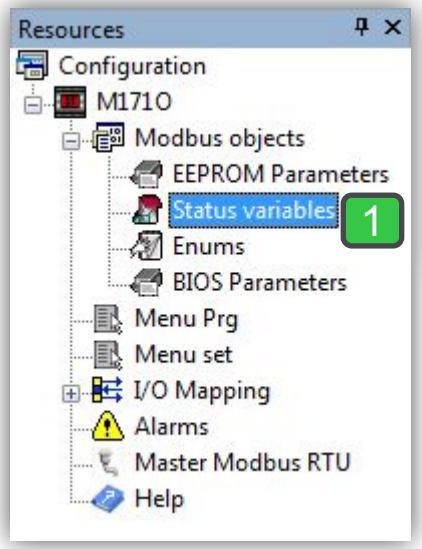
Add Remove Recalc

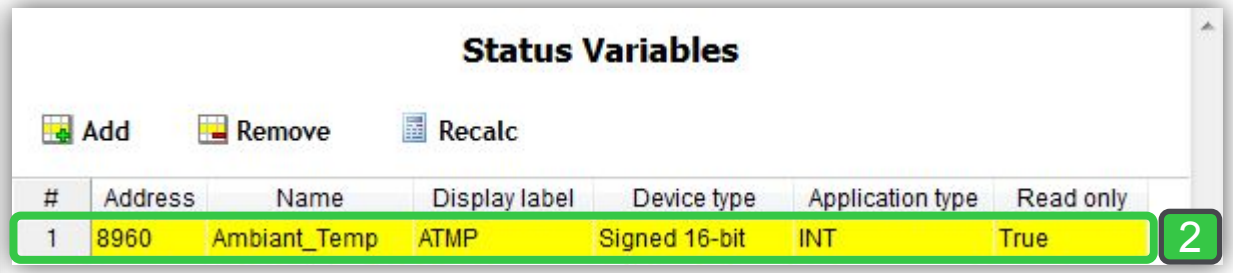
#	Address	Name	Display label	Device type	Application type	Default value	Min	Max	Unit	Format	AccessLevel
1	16384	SetPoint	SetP	Signed 16-bit	INT	180	120	300	°C	XXX.Y	Always visible
2	16385	Delta	Dlta	Signed 16-bit	INT	25	10	50	°C	XXX.Y	Always visible

2

Status Variable definition/list

Slave side

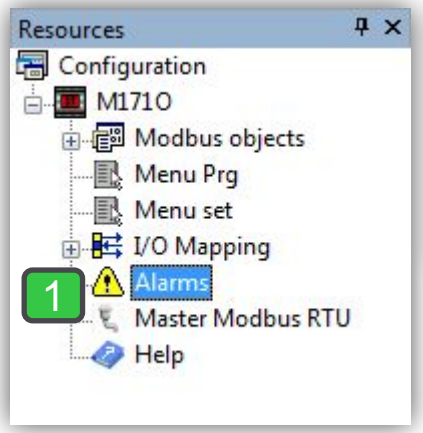




Note:

Alarm definition/list

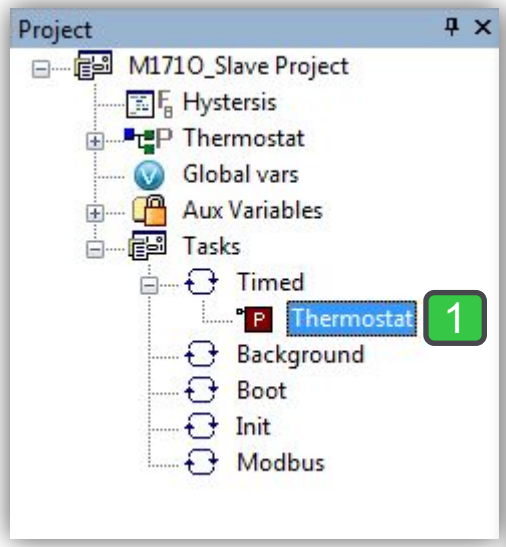
Slave side

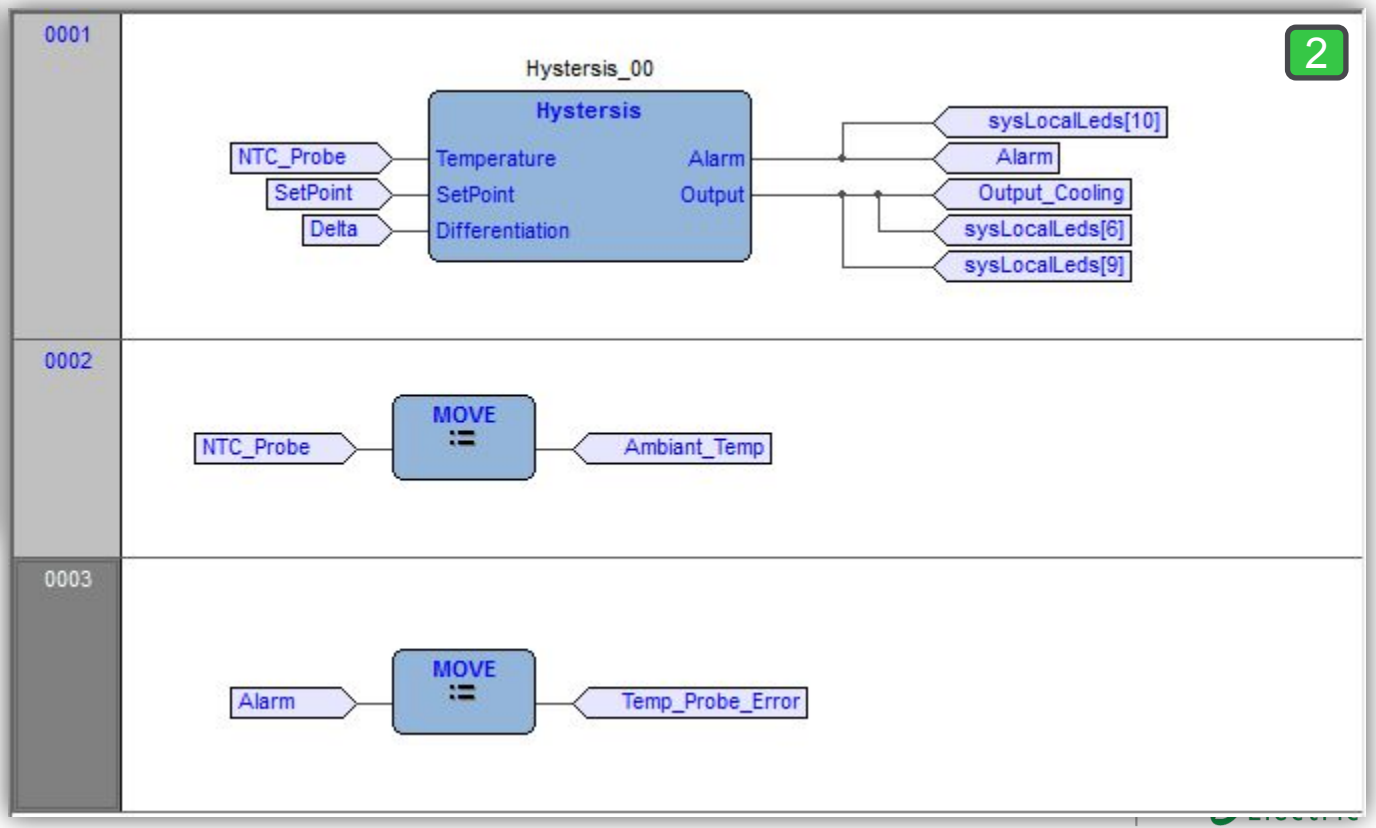


Note:
Alarm list definition advantages are as below:
1. Automatic foldering when an alarm raised and shrink the list pack when it disappeared
2. Automatic launching of hazard sign/icon in front face of product (system LED)

Thermostat Program

Slave side

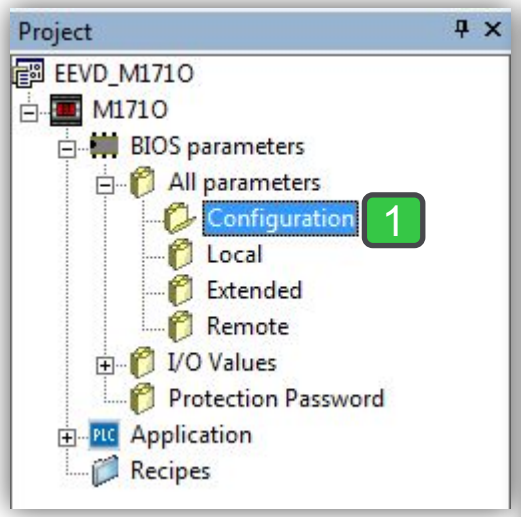




Note:

Modbus Protocol Properties

Slave side



2 Configuration

Address	Name	Value	Um	Default	Min	Max	Description
53276	CF32	1=Even	num	1=Even	1	3	Modbus parity protocol
53274	CF30	1	num	1	1	255	Modbus protocol controller address
15636	Par_POLI	0	num	0	0	65535	Polycarbonate code
15715	Ui26	350	4ms	350	0	999	Key hold time to enable function
53456	CF50	1=Present	num	1=Present	0	1	RTC present
15640	CF61	0	num	0	0	999	Customer code 2
15639	CF60	0	num	0	0	999	Customer code 1
15744	Ui27	1	num	1	0	255	Installation engineer password
53275	CF31	3=9600	num	3=9600	0	7	Modbus baud rate protocol
15745	Ui28	2	num	2	0	255	Manufacturer password
53273	CF21	0	num	0	0	14	Protocol controller family
53272	CF20	0	num	0	0	14	Protocol controller address
53265	CF01	1	num	1	0	1	Select COM1 protocol

Note:

Target and SoMachineHVAC

Parameters needed for correct connection between the **M171O targets and** SoMachineHVAC

5. Set them as default value

parameter	description	values	default	visibility	notes
1 **CF30**	Modbus protocol controller address	1...255	1	3	Check that the set values correspond to those defined by the panel **Communication > Settings > Properties**
CF31**	Modbus protocol baud rate	0,1, 2 = not used 3 = 9600 baud 4 = 19200 baud 5 = 38400 baud 6 = 57600 baud 7 = 115200 baud	3	3	
3 **CF32**	Modbus protocol controller parity	1= EVEN 2 = NONE 3 = ODD	1	3	
*COM1 = TTL / RS485 (/S models only): cannot be used simultaneously					
**CF31			5=38400 baud (RS485: not supported) 6=57600 baud (RS485: non supported) 7=115200 baud (RS485: non supported)		

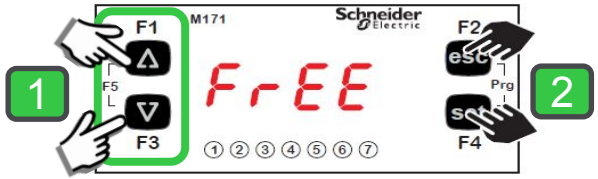
4. Maximum supported baud rate is 19200

Life Is On Schneider Electric

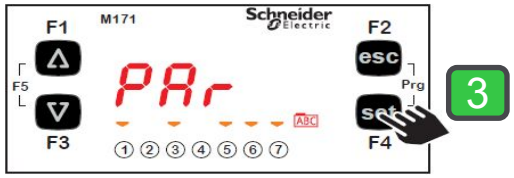
Customize M171O Baud Rate

M171O parameters in the CF folder manages the connection between the target and Studio

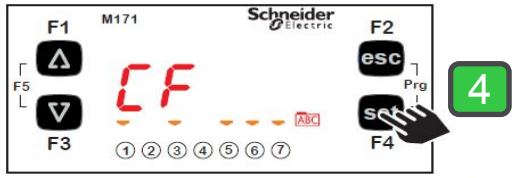
If the target is “empty”, i.e. there is no IEC application on the device, M171O will display the message FrEE, otherwise fundamental state is displayed (Press F5 to switch to FrEE menu)



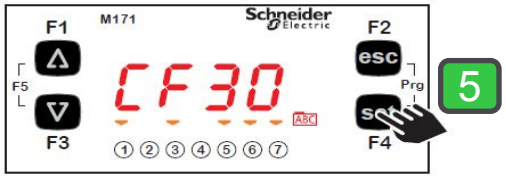
To view the parameter menu, press the Esc and Set keys at the same time. This will open the PAr menu.



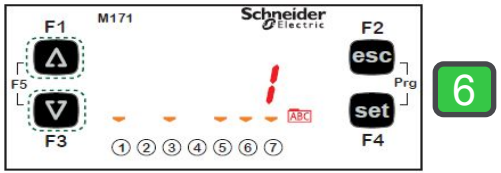
The parameters menu PAr contains all controller folders. Press the set key to view folders.



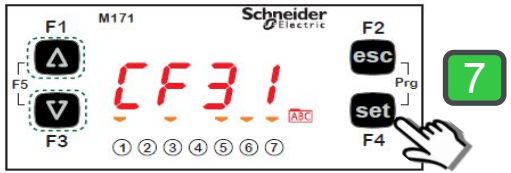
The first folder shown is the CF configuration folder. Press the set key to view the folder parameters.



The first parameter shown is CF30. To view the value of the parameter press the set key.



Use the UP and DOWN keys to change the value if necessary. To confirm the value press the set key. To exit press Esc



Use the UP and DOWN keys to scroll the other parameters and repeat the procedure to view the values and - if necessary - edit them.

M171O Controller

Modbus Serial line Configuration, **Master Side**

M171O Modbus Limitations

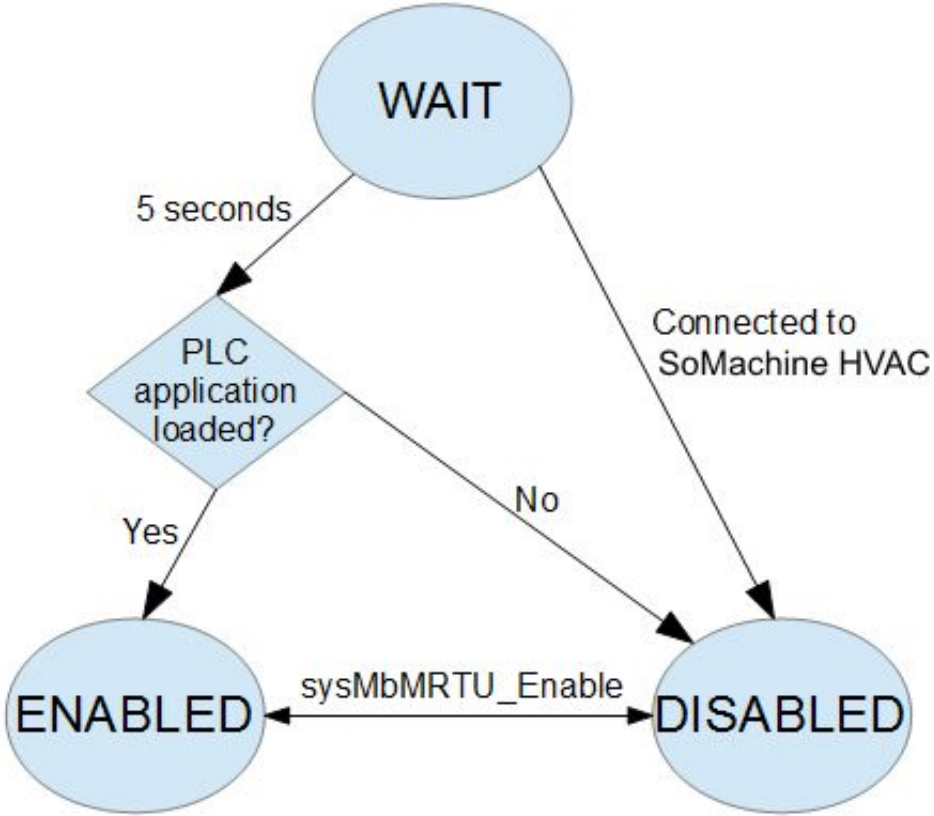
Master Side

- Maximum number of nodes 127
- The maximum number of Modbus message is related of the available RAM in the device.
- Each message is a Function block that use around 64 byte of the RAM Memory.
- Each Modbus message can link up to 8 StatusVariable/Parameter/

Modbus RTU master enable status

Master Side

Since the communication port used by the Modbus RTU master is the same used by SoMachineHVAC to connect to a Modicon M171 programmable controller, you cannot connect to the device if the Modbus RTU master is running.



Modbus Master Disabling methods,

Master Side

1. Temporarily:

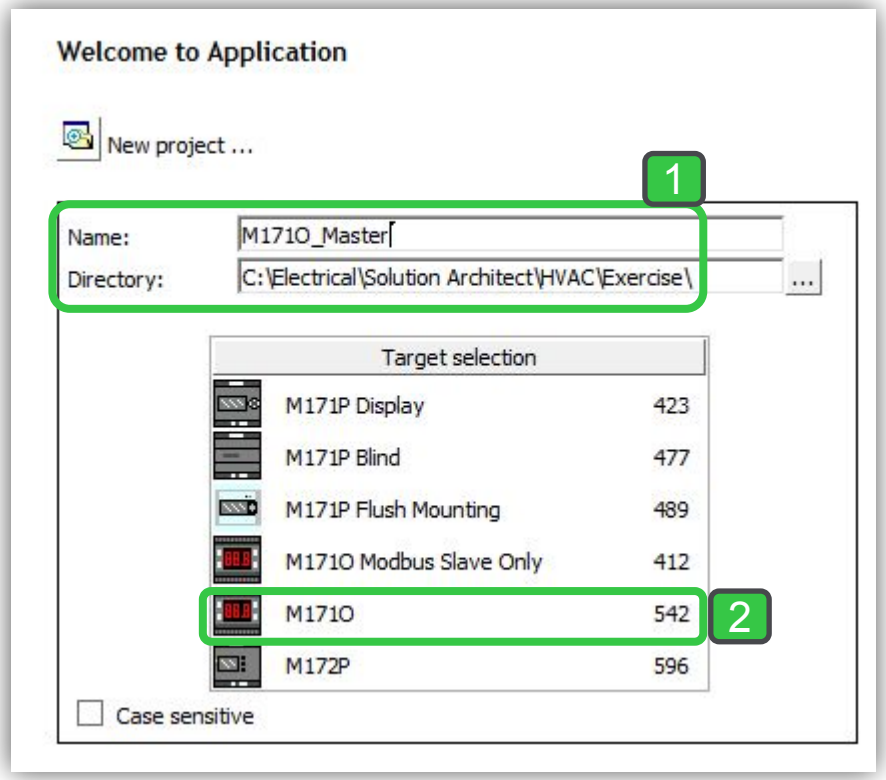
Delayed start: after the controller is powered on, it waits for approximately 5 seconds for a connection with the development environment to be established, before loading the PLC application and enabling the Modbus RTU master; if the connection is established, the PLC application is loaded but the Modbus RTU master is not enabled;

2. Permanently:

Run-time disable: Modbus RTU master can be dynamically disabled and enabled again by the PLC application (through a call to the target FUNCTION sysMbMRTU_Enable), to allow SoMachine HVAC - Application's debuggers to attach to the target device even if the PLC application has already been started.

Creating New project

Master side, Controller selection



Note: Only mask version 542 can supports Modbus Master Functionality

Modbus Master

sysMbMRTU_Enable

Enable/Disable Modbus master: that allows to dynamically switch to programming/debug mode

Type : FUNCTION

Code type : EMBEDDED

Input vars num: **1**

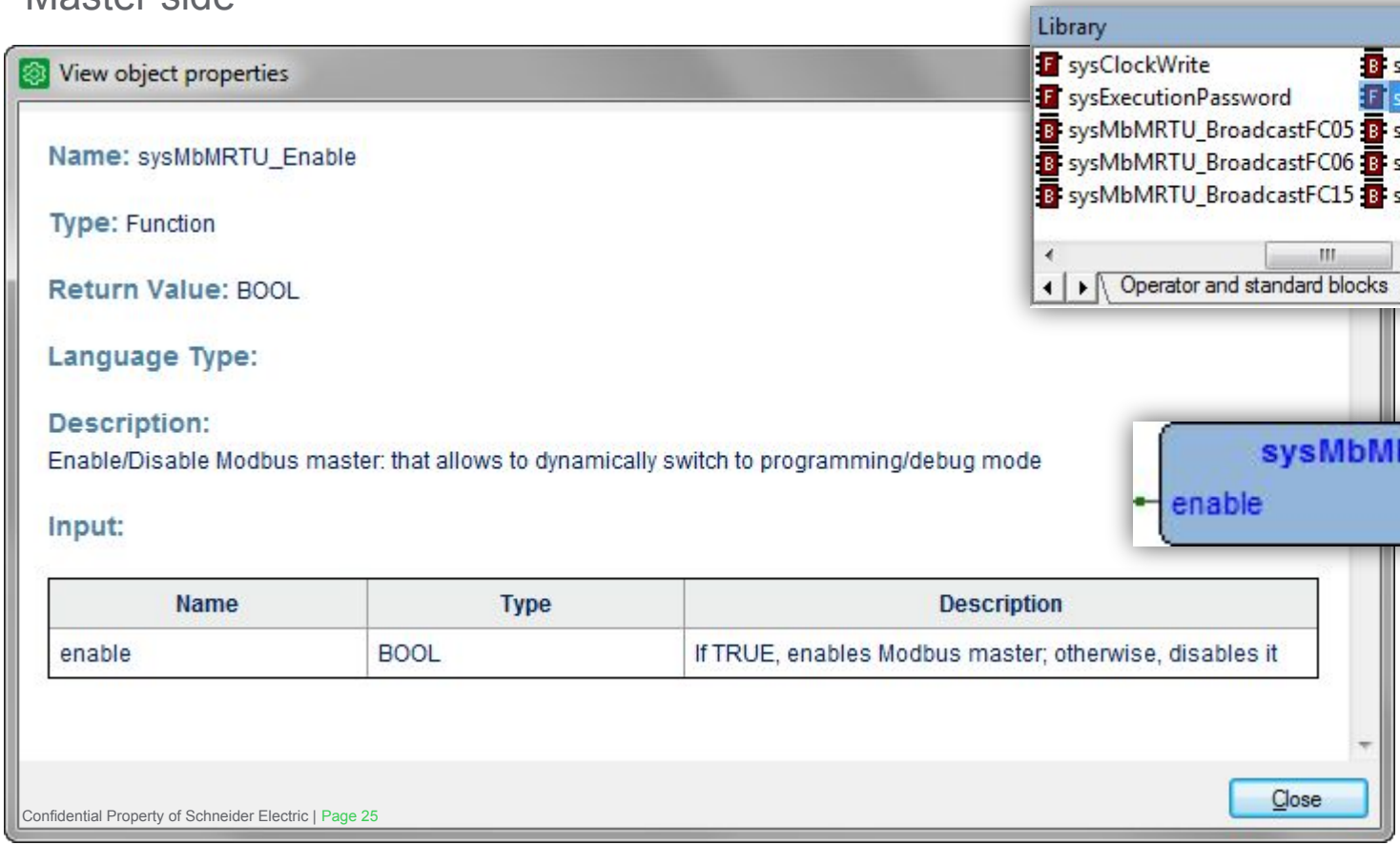
enable : BOOL (* If TRUE, enables Modbus master; otherwise, disables it *)

Result type : BOOL

Note: By default Modbus Master Functionality is deactivated that's why it is needed to activate it by sysMbMRTU_Enable function.

Modbus Master Enable/Disable

Master side



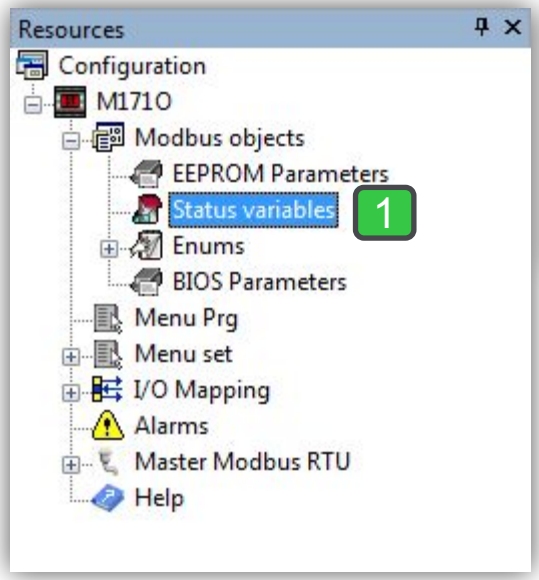
1

2

Note:

Status Variable Defenition

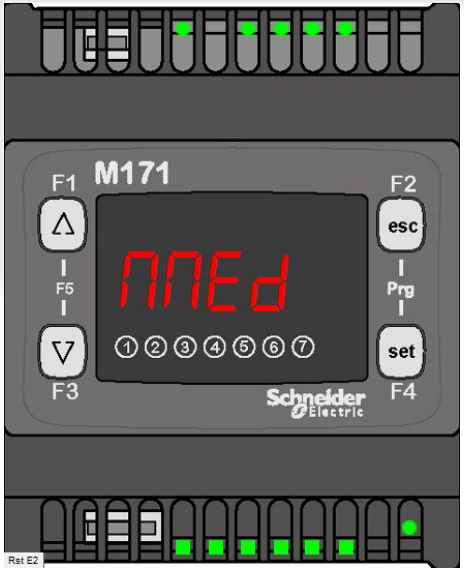
Master side, to Activate/Deactivate Modbus Master Functionality



Status Variables

Add Remove Recalc 2

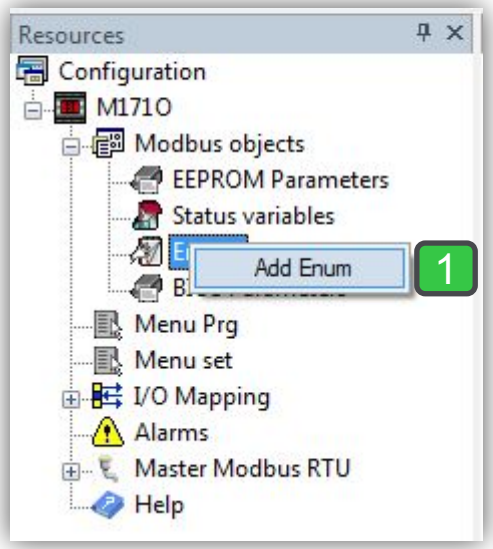
#	Address	Name	Display label	Device type	Application type	Default value	Read only
1	8960	Enable_Disable	MMED	Boolean	BOOL	False	False

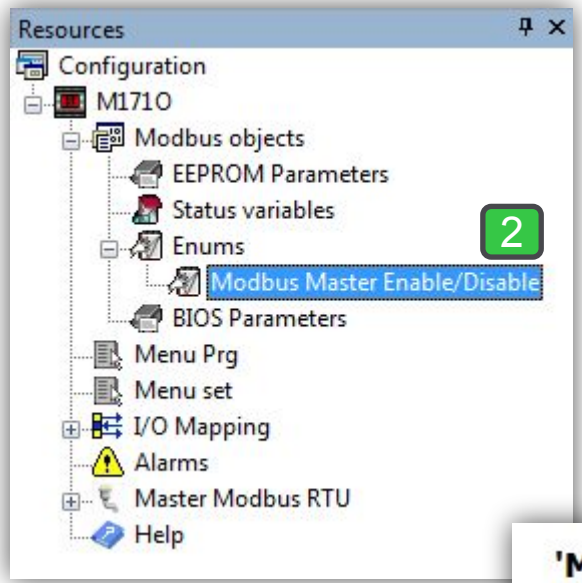


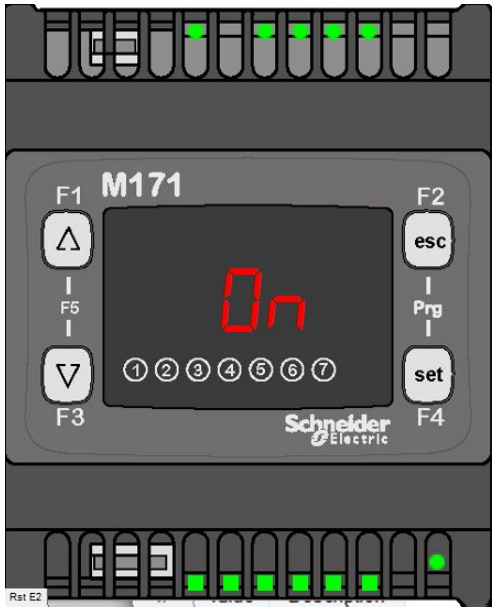
Note:

Enable/Disable Modbus Master by Enums

Master side



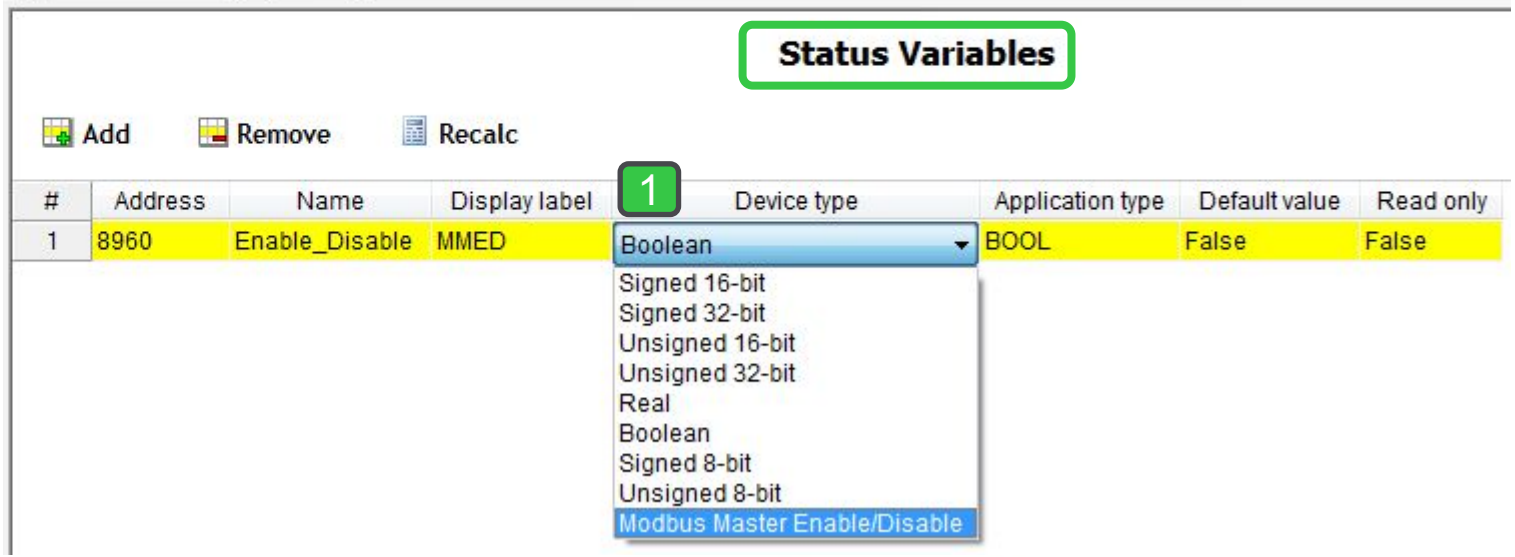


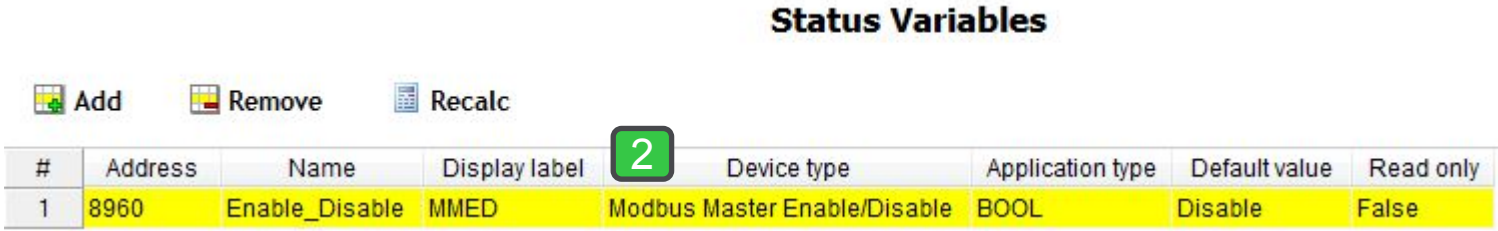


Note:

Set the Enumerator as Device type

Master side

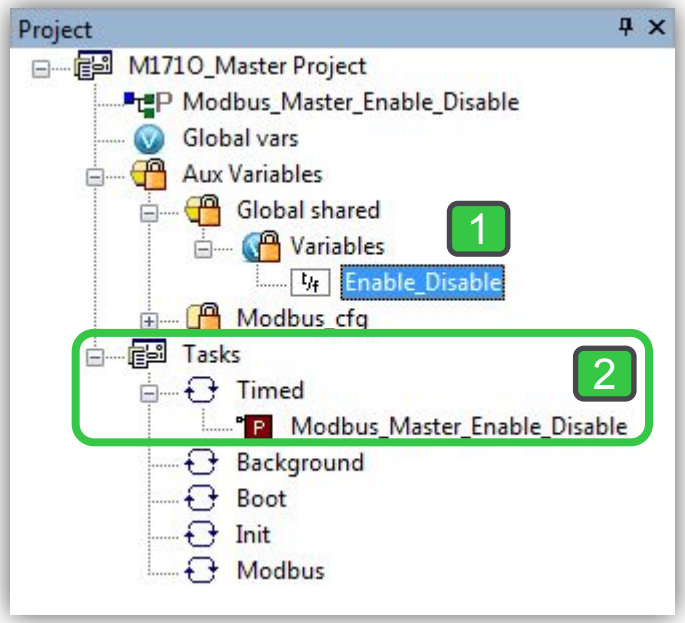


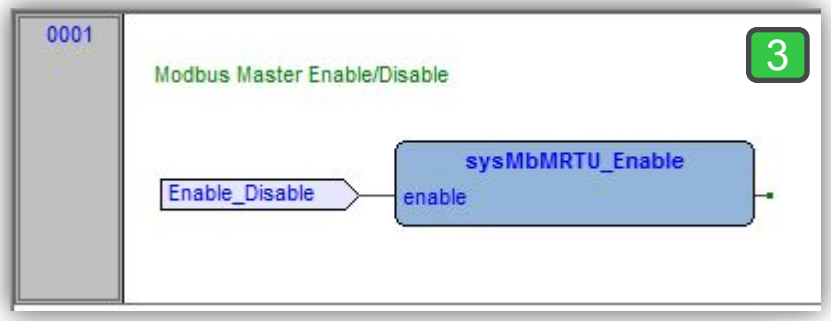


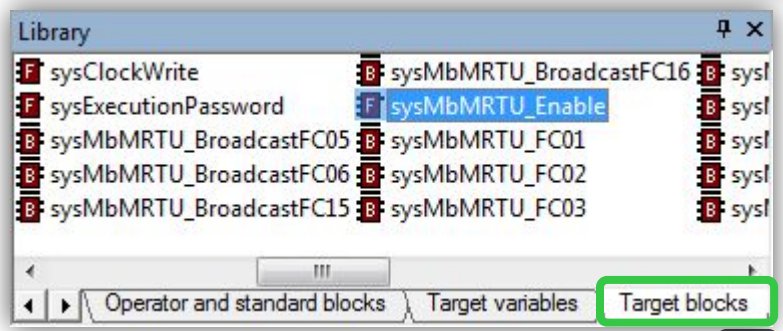
Note:

Enable/Disable Modbus Master Program

Master side



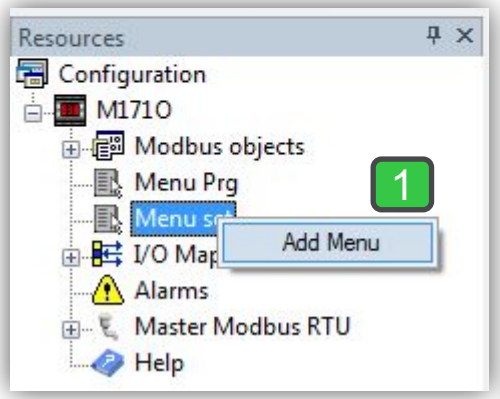


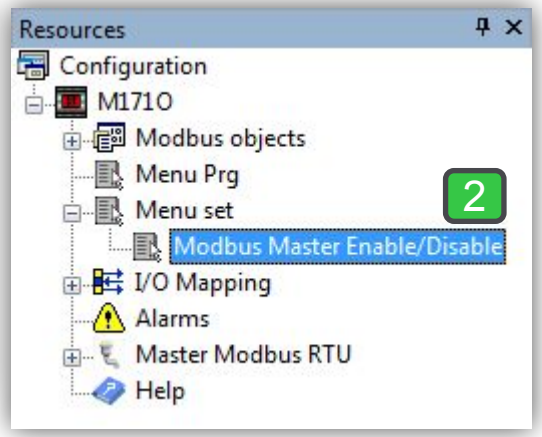


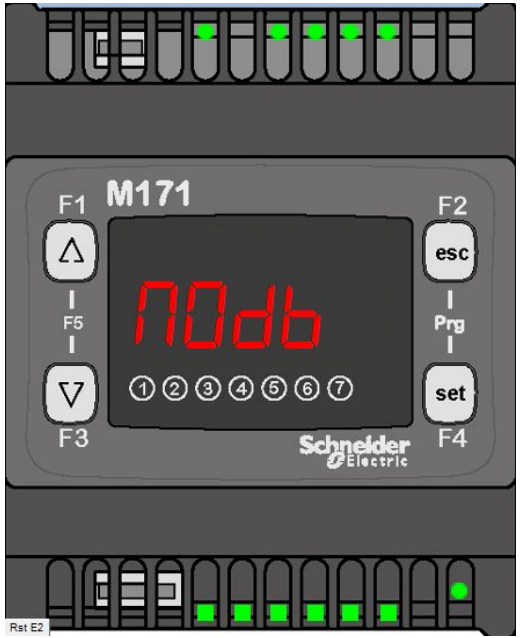
Note:

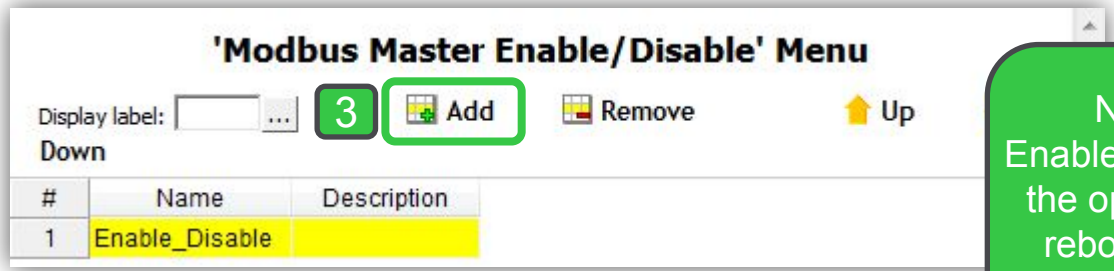
Set Menu Creation

Master side, to access Enable/Disable from dashboard





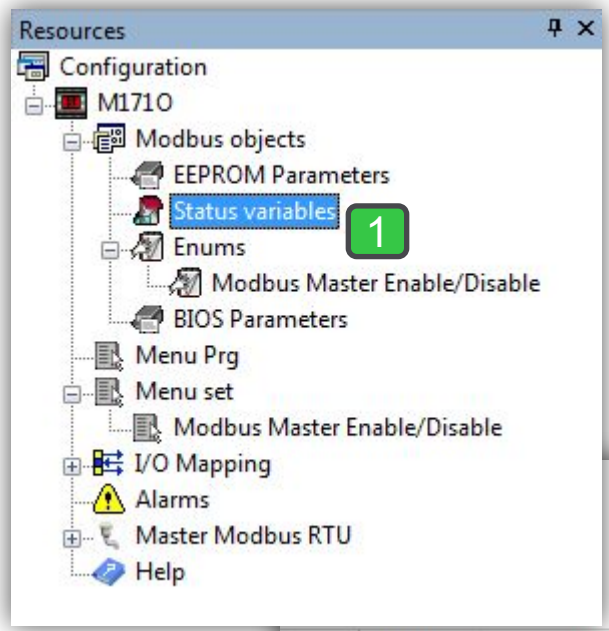




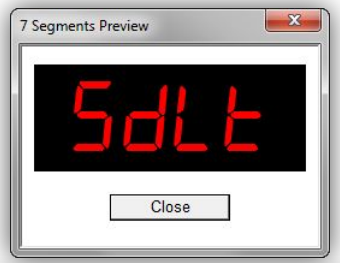
Note: If you do create a menu to access Enable/Disable from front face of product you have the option of Modbus Master deactivation during rebooting the controller (in 5 sec) to be able to connect to controller via SoMachineHVAC.

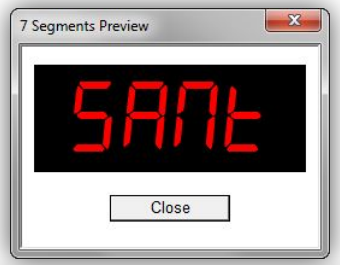
Status Variable Definition

Master side



Note: As far as Set point & Delta are defined as EEPROPM parameters in slave side no need to keep their values again in master side and just reading/writing them via status variable is enough.





3

Status Variables

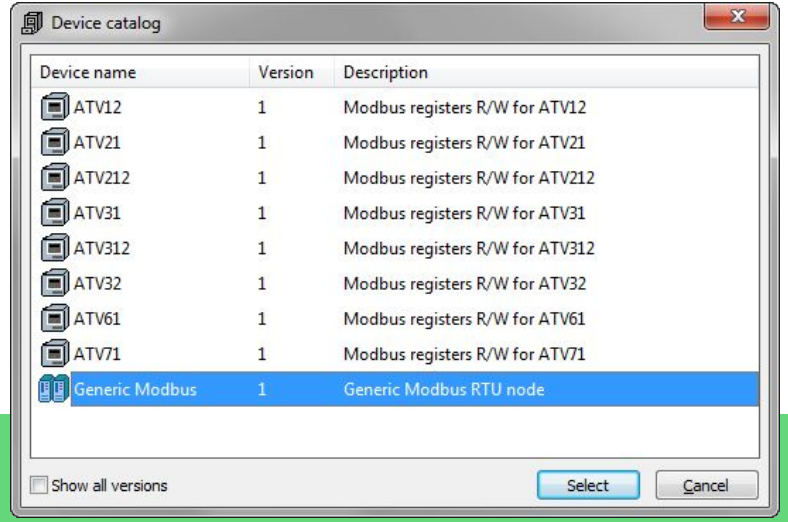
Recalc

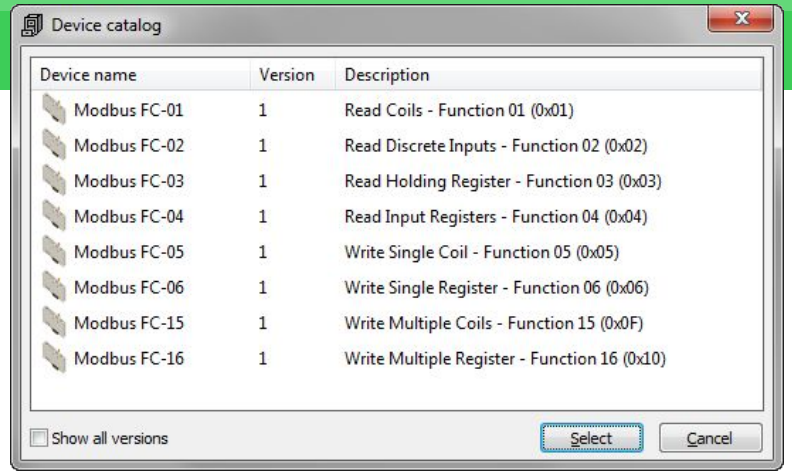
#	Address	Name	Display label	Device type	Application type	Default value	Read only
1	8960	Enable_Disable	MMED	Modbus Master Enable/Disable	BOOL	Disable	False
2	8961	Slave_Setpoint	SSEP	Signed 16-bit	INT		False
3	8962	Slave_Delta	SDLT	Signed 16-bit	INT		False
4	8963	Slave_Ambiant_Temp	SAMT	Signed 16-bit	INT		False

2

1st Method: Generic Modbus RTU

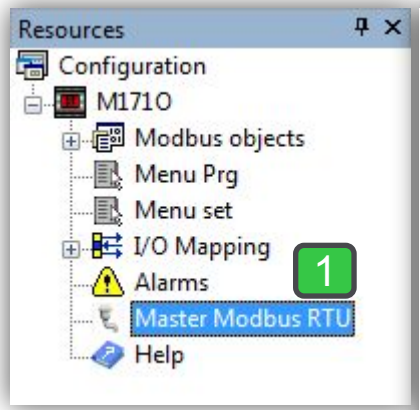
By variable assignment

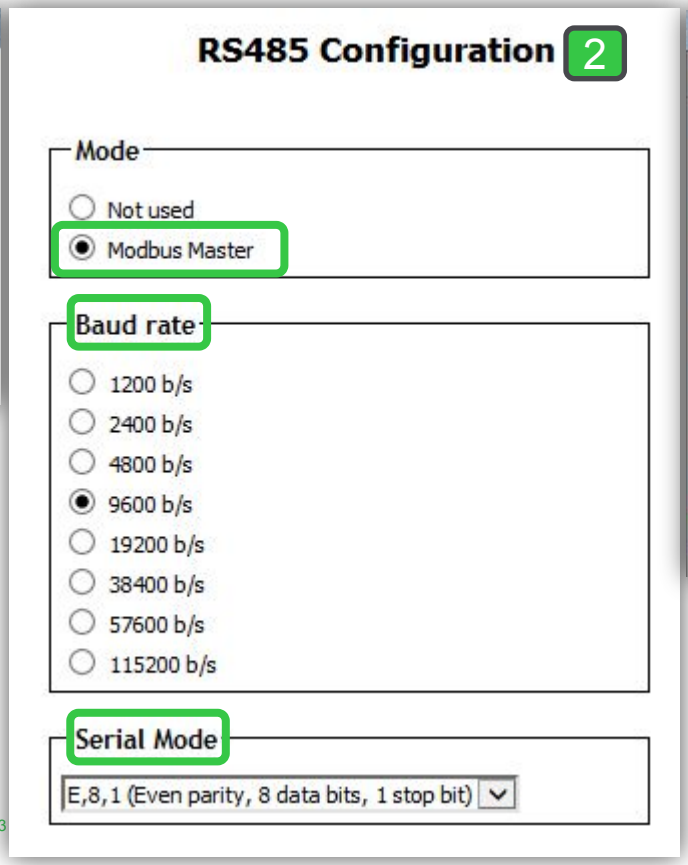


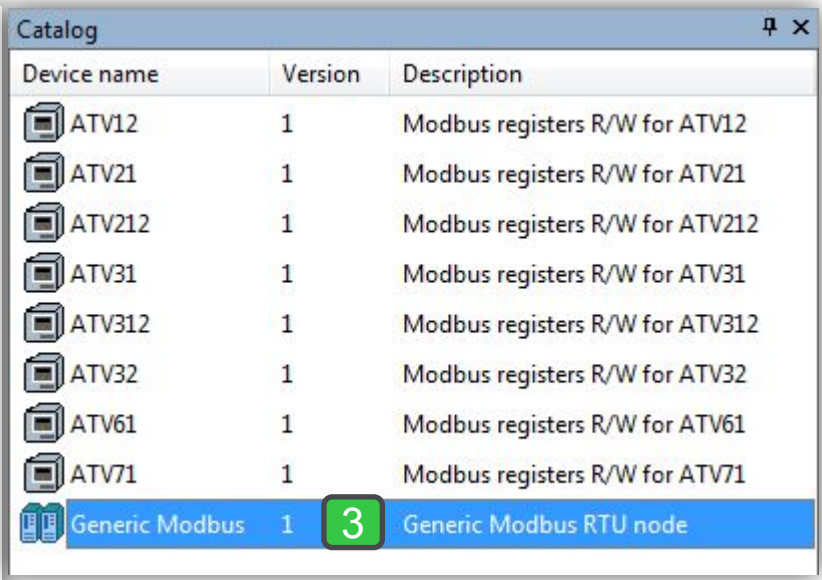


Modbus Master, RS485 Configuration

Master side



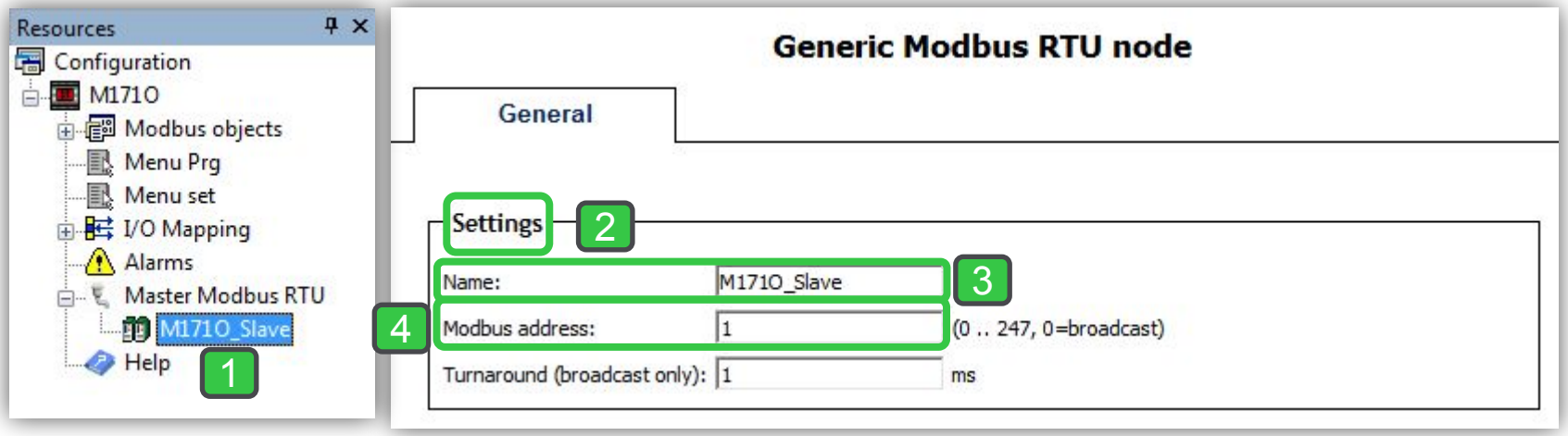




Note:

Generic Modbus RTU

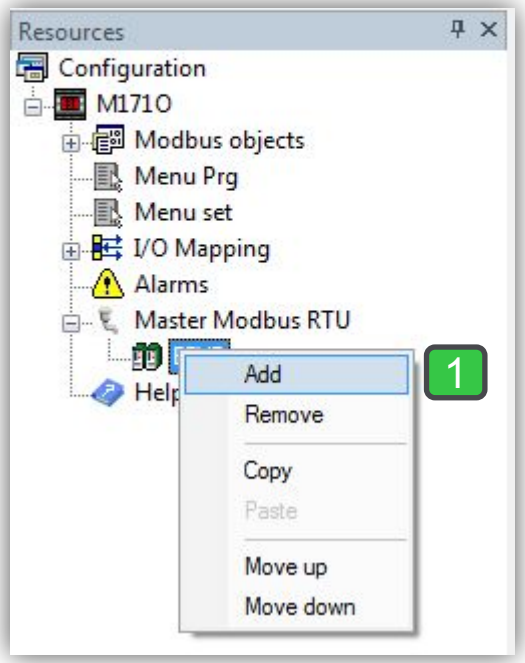
Master side

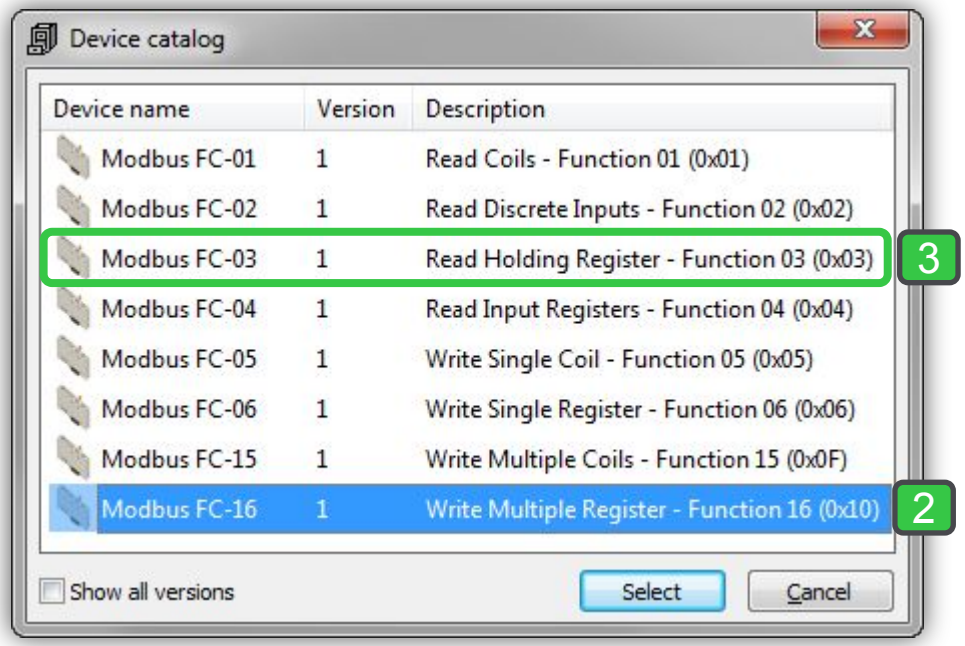


3. Rename it to a meaning full name
4.Se the slave's Modbus address

Read and Write Function Codes

Master side, Read & Write

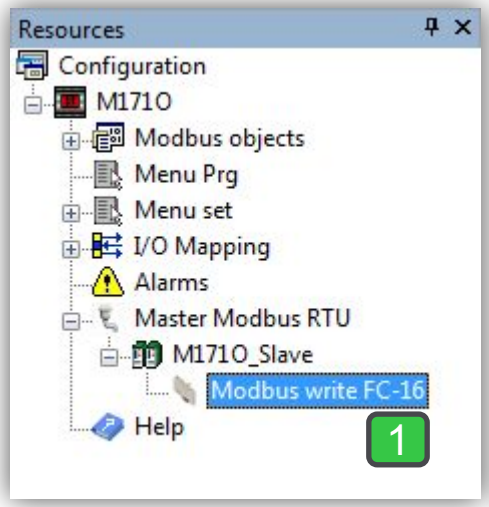


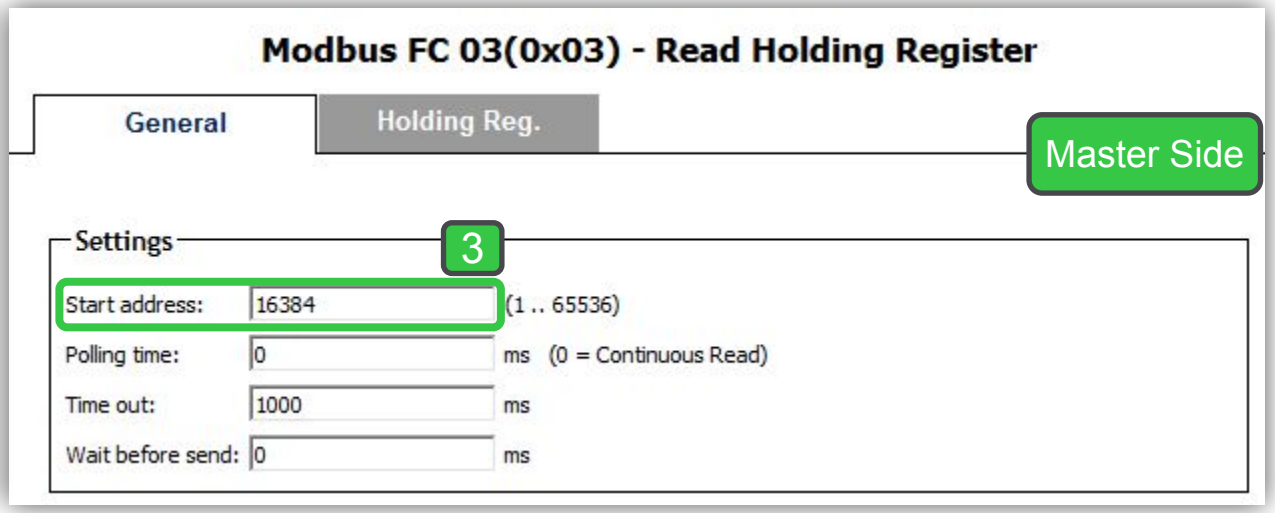


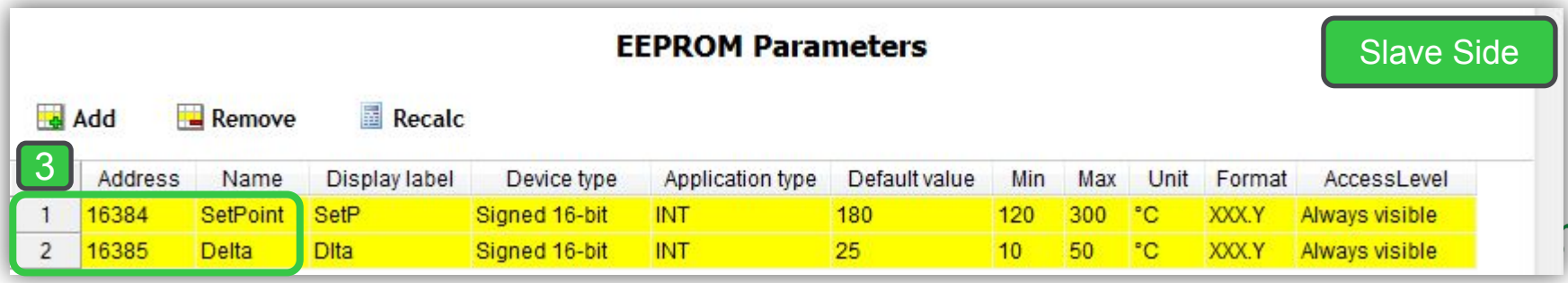
2 & 3. Drag and drop the desired function code/s into the Modbus Master RTU link

Writing Multiple Register General Settings

Master side, Address matching



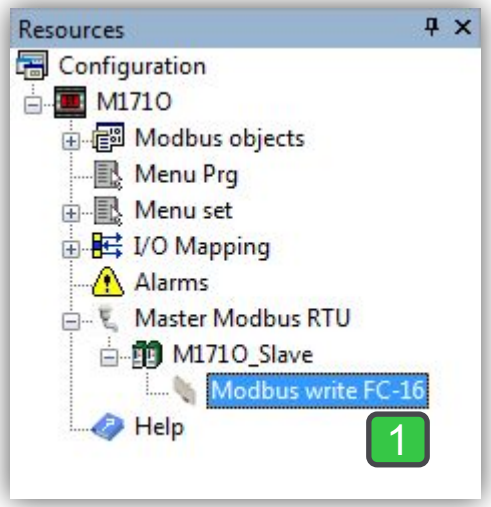


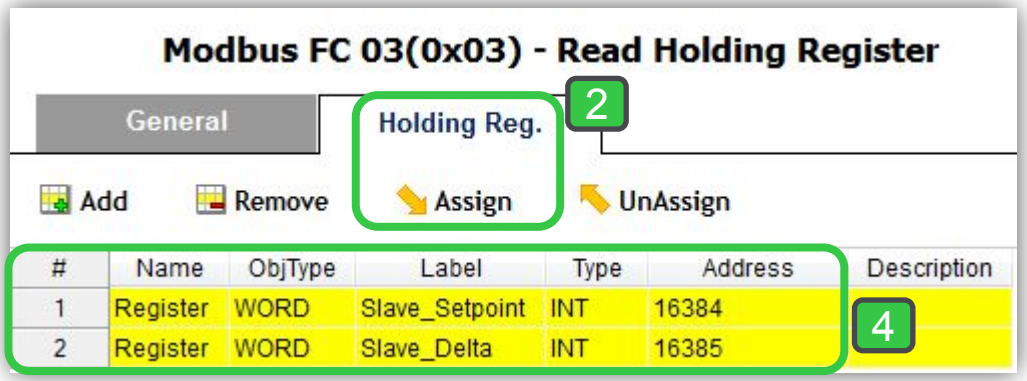


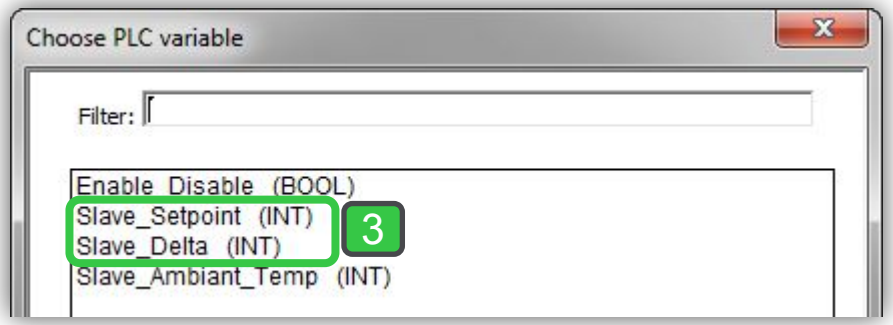
	Address	Name	Display label	Device type	Application type	Default value	Min	Max	Unit	Format	AccessLevel
1	16384	SetPoint	SetP	Signed 16-bit	INT	180	120	300	°C	XXX.Y	Always visible
2	16385	Delta	Dlta	Signed 16-bit	INT	25	10	50	°C	XXX.Y	Always visible

Variable Assignment/Witre Function Code

Master side, Assignment



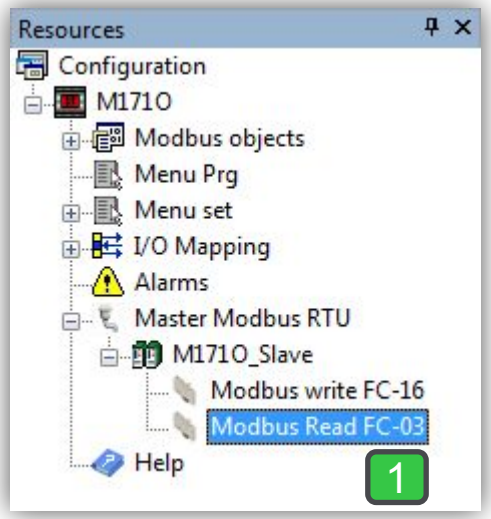


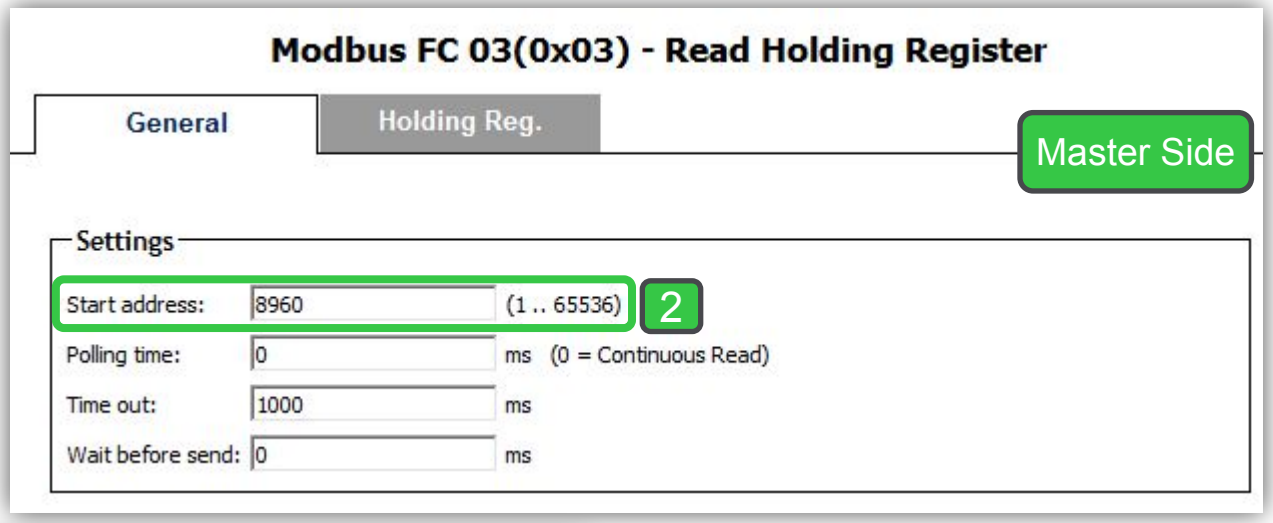


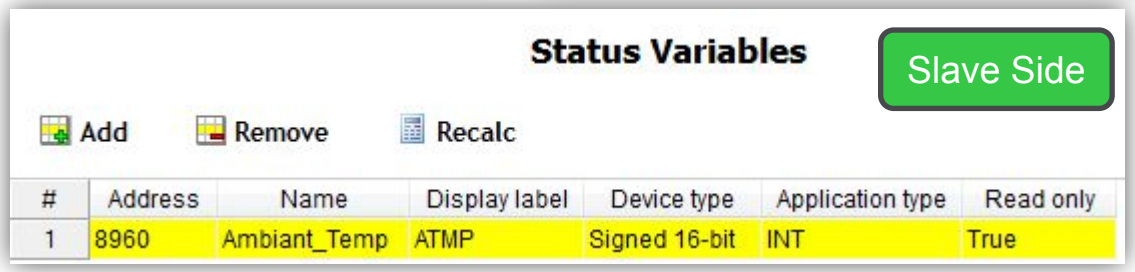
Note:

Read Holding Register General Settings

Master side, Address matching



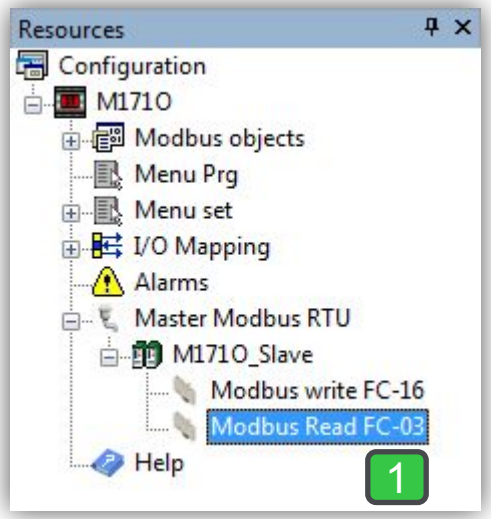


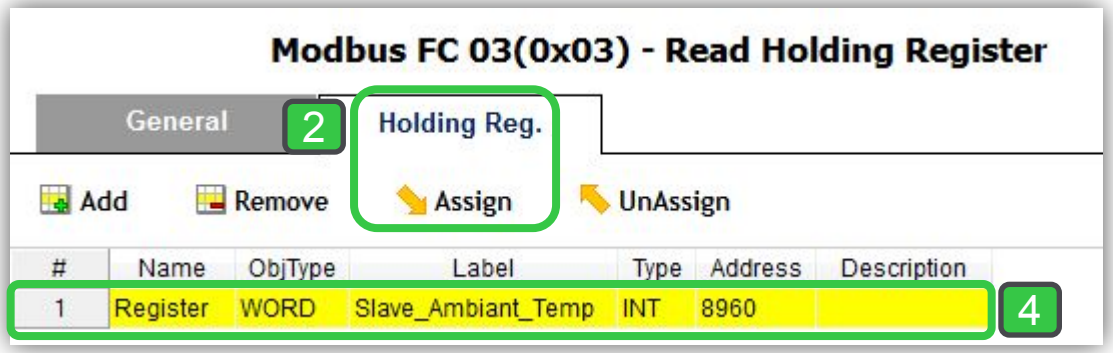


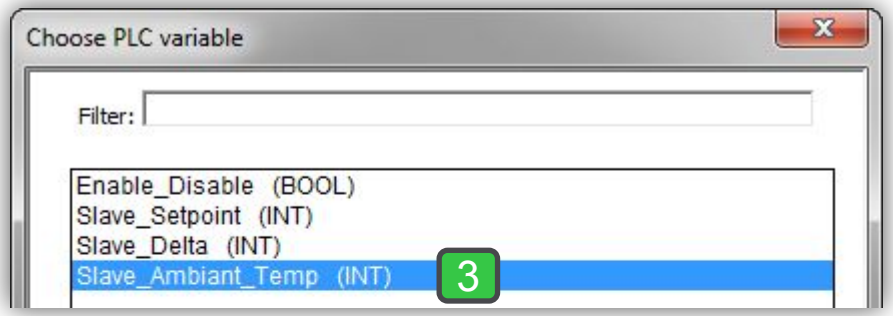
Note:

Variable Assignment/Read Function Code

Master side, Assignment



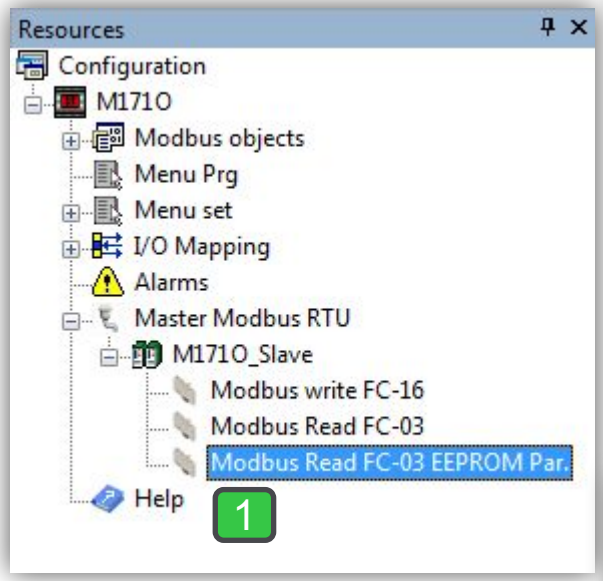


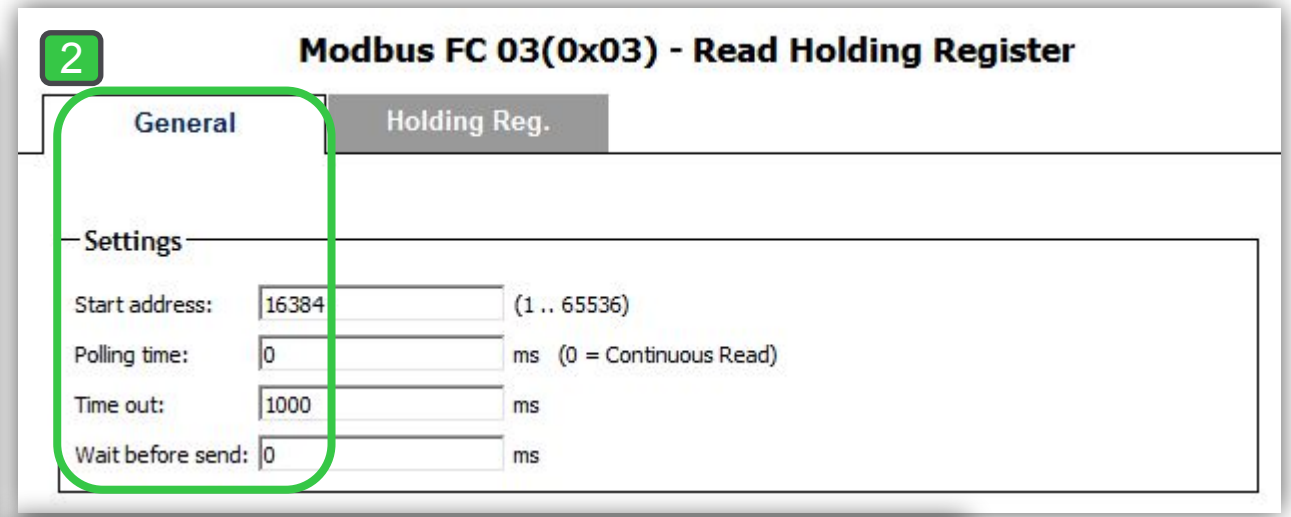


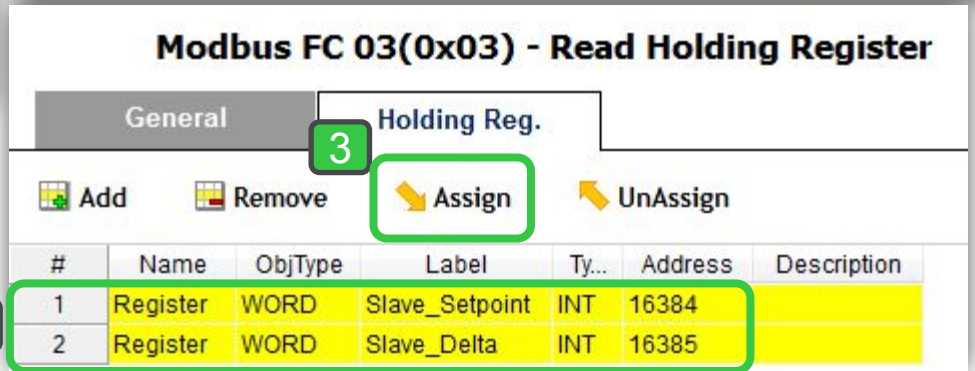
Note:

Read Holding Register General Settings

Master side, address matching & assignment



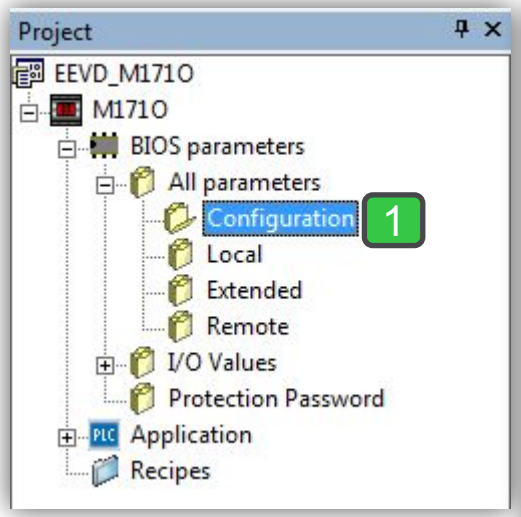




Note:

Modbus Protocol Properties

Master side, via Device tool



2 **Configuration**

Address	Name	Value	Um	Default	Min	Max	Description
53276	CF32	1=Even	num	1=Even	1	3	Modbus parity protocol
53274	CF30	1	num	1	1	255	Modbus protocol controller address
15636	Par_POLI	0	num	0	0	65535	Polycarbonate code
15715	Ui26	350	4ms	350	0	999	Key hold time to enable function
53456	CF50	1=Present	num	1=Present	0	1	RTC present
15640	CF61	0	num	0	0	999	Customer code 2
15639	CF60	0	num	0	0	999	Customer code 1
15744	Ui27	1	num	1	0	255	Installation engineer password
53275	CF31	3=9600	num	3=9600	0	7	Modbus baud rate protocol
15745	Ui28	2	num	2	0	255	Manufacturer password
53273	CF21	0	num	0	0	14	Protocol controller family
53272	CF20	0	num	0	0	14	Protocol controller address
53265	CF01	1	num	1	0	1	Select COM1 protocol

Note:

Target and SoMachineHVAC

Parameters needed for correct connection between the **M171O targets and** SoMachineHVAC

5. Set them as default value

parameter	description	values	default	visibility	notes
1 **CF30**	Modbus protocol controller address	1...255	1	3	Check that the set values correspond to those defined by the panel **Communication > Settings > Properties**
CF31**	Modbus protocol baud rate	0,1, 2 = not used 3 = 9600 baud 4 = 19200 baud 5 = 38400 baud 6 = 57600 baud 7 = 115200 baud	3	3	
3 **CF32**	Modbus protocol controller parity	1= EVEN 2 = NONE 3 = ODD	1	3	
*COM1 = TTL / RS485 (/S models only): cannot be used simultaneously					
**CF31			5=38400 baud (RS485: not supported) 6=57600 baud (RS485: non supported) 7=115200 baud (RS485: non supported)		

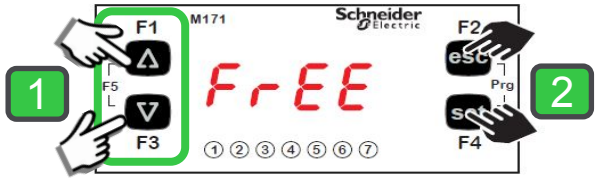
4. Maximum supported baud rate is 19200

Life Is On Schneider Electric

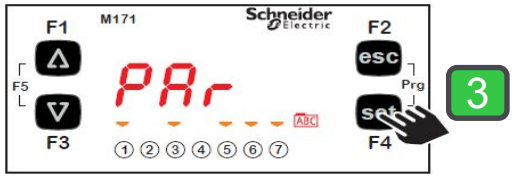
Customize M171O Baud Rate

M171O parameters in the CF folder manages the connection between the target and Studio

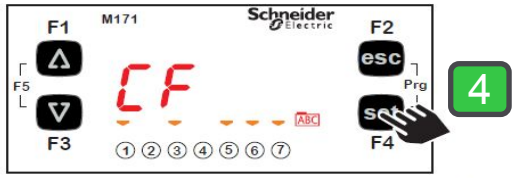
If the target is “empty”, i.e. there is no IEC application on the device, M171O will display the message FrEE, otherwise fundamental state is displayed (Press F5 to switch to FrEE menu)



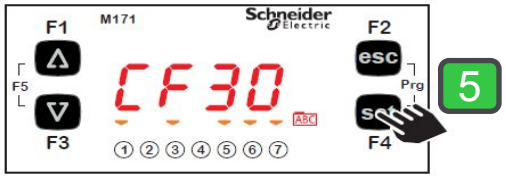
To view the parameter menu, press the Esc and Set keys at the same time. This will open the PAr menu.



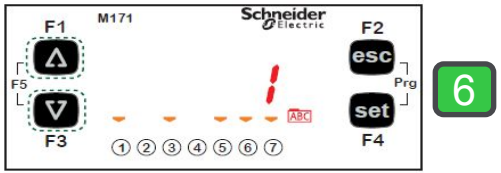
The parameters menu PAr contains all controller folders. Press the set key to view folders.



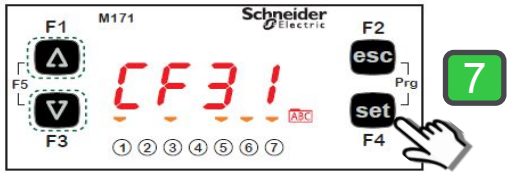
The first folder shown is the CF configuration folder. Press the set key to view the folder parameters.



The first parameter shown is CF30. To view the value of the parameter press the set key.



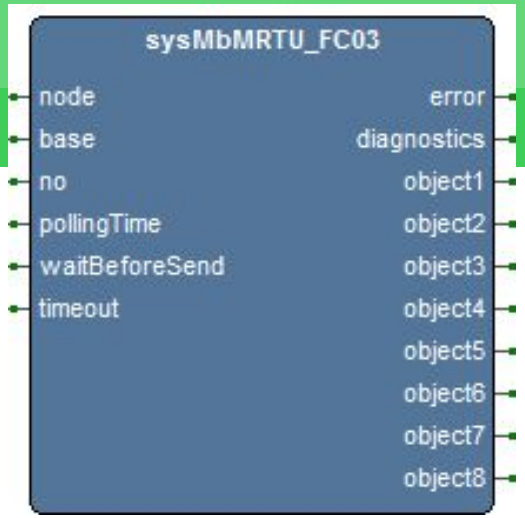
Use the UP and DOWN keys to change the value if necessary. To confirm the value press the set key. To exit press Esc

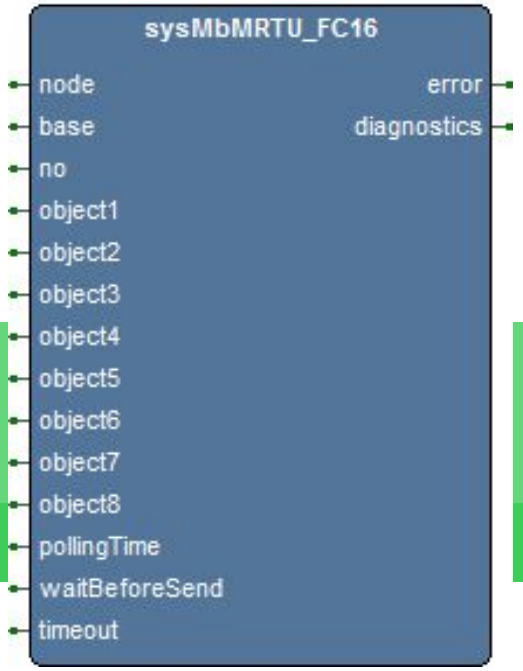


Use the UP and DOWN keys to scroll the other parameters and repeat the procedure to view the values and - if necessary - edit them.

2nd Method: Target Blocks

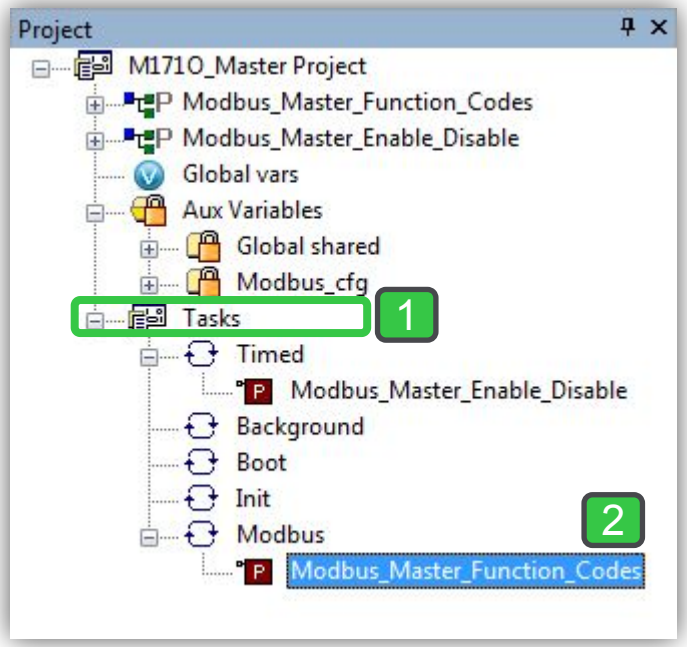
By Function Block Configuration





Modbus Task

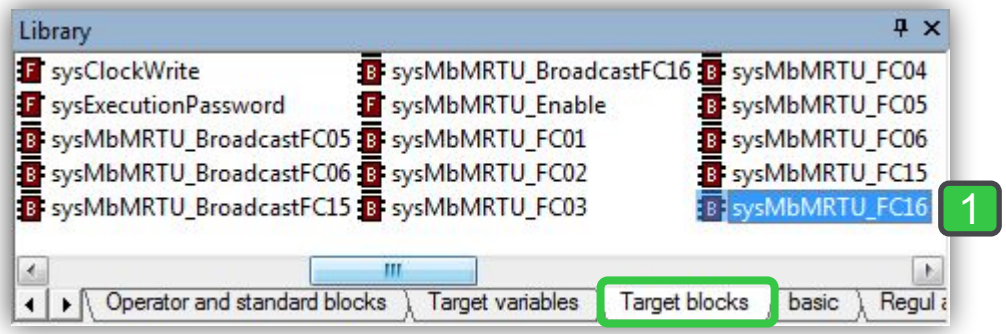
Master side

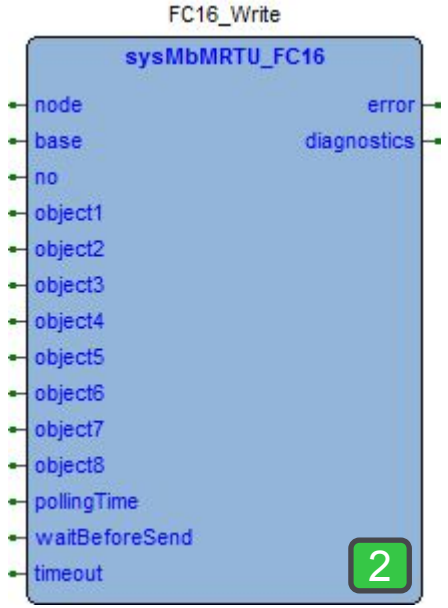


Note:
Dedicated PLC task (Modbus task): operations on the bus do not affect the PLC Timed task and have limited impact on lower-priority tasks, since they are carried on asynchronously.

sysMbMRTU write function code, FC16

Master side, library & Output Pin-Out





Output:

Name	Type	Description
error	BOOL	TRUE=error occurred,FALSE=no error
diagnostics	MbMRTUDiagnostics	Message diagnostics

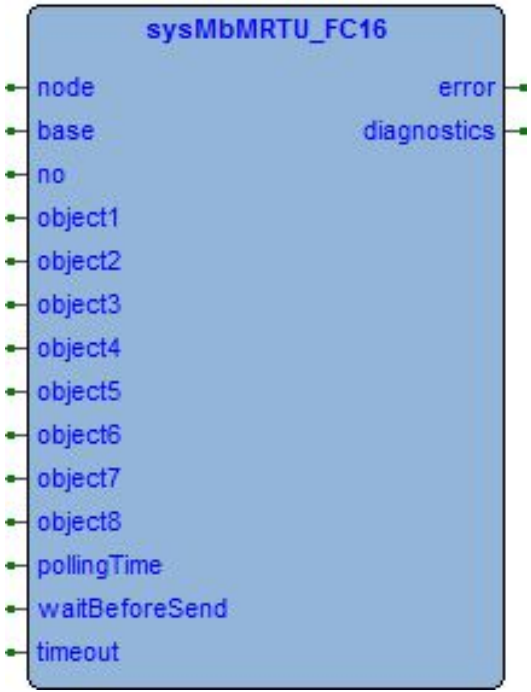
Note: This function block can be used in the Modbus task only.

sysMbMRTU write function code, FC16

Master side, Input Pin-Out

Input:

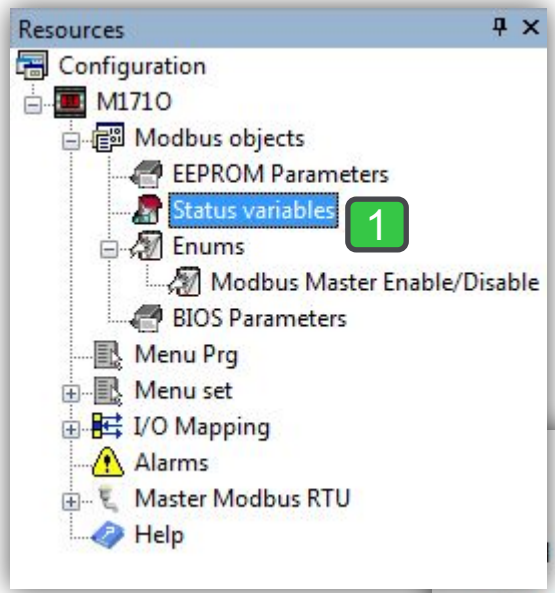
Name	Type	Description
node	@MbMRTUNode	Slave node
base	UINT	Address of the first Holding Register to write
no	UINT	Number of contiguous Holding Registers to write
object1	INT	1st Holding Register value
object2	INT	2nd Holding Register value
object3	INT	3rd Holding Register value
object4	INT	4th Holding Register value
object5	INT	5th Holding Register value
object6	INT	6th Holding Register value
object7	INT	7th Holding Register value
object8	INT	8th Holding Register value
pollingTime	UINT	Polling time [ms]. 0=write on variation
waitBeforeSend	UINT	Time to wait before sending the message [ms]
timeout	UINT	Timeout [ms]



Note:

Modbus Communication Error Detection

Master Side, Status Variable declaration



Status Variables

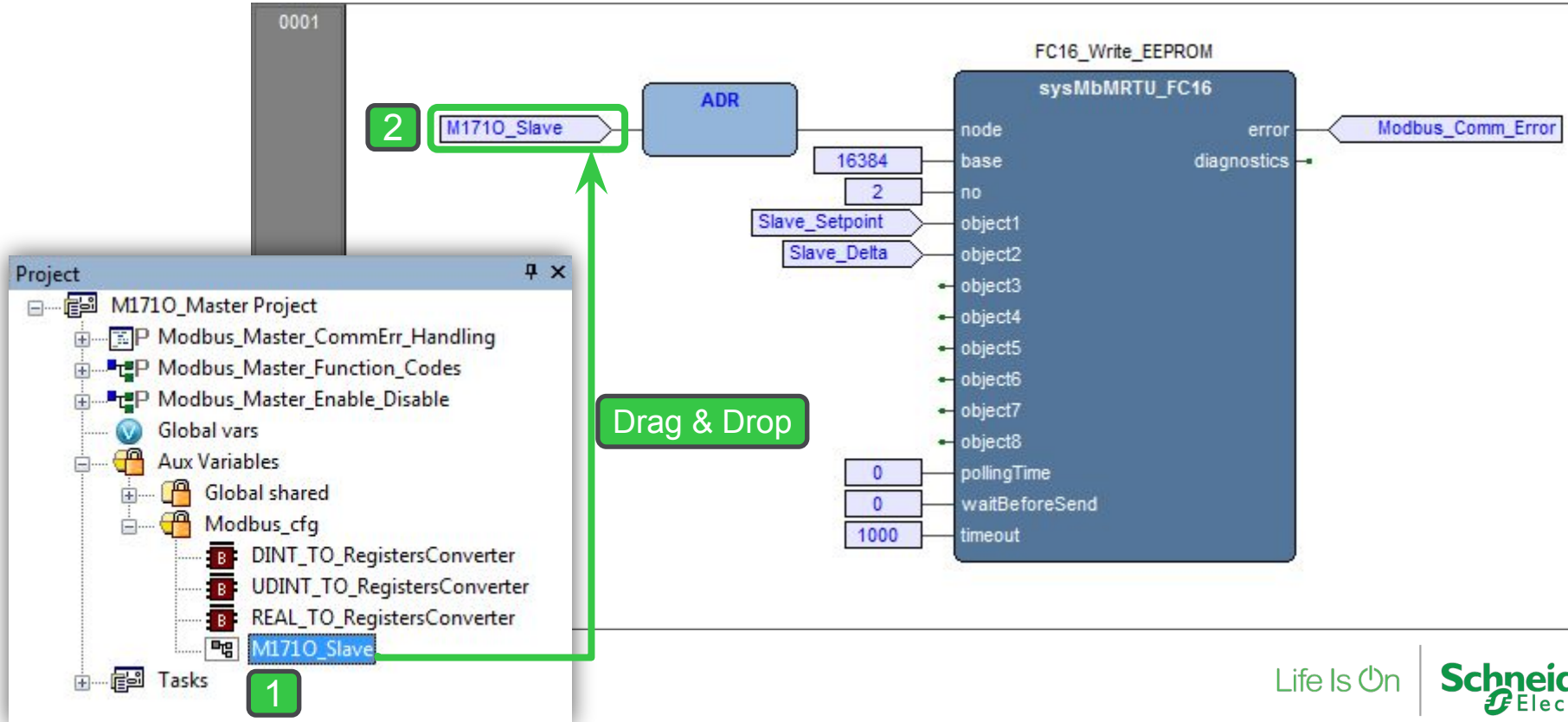
Remove Recalc

#	Address	Name	Display label	Device type	Application type	Format	Read only
1	8960	Enable_Disable	MMED	Modbus Master Ei	BOOL		False
2	8961	Slave_Setpoint	SSEP	Signed 16-bit	INT	XXX.Y	False
3	8962	Slave_Delta	SDLT	Signed 16-bit	INT	XXX.Y	False
4	8963	Slave_Ambiant_Temp	SAMT	Signed 16-bit	INT	XXX.Y	False
5	8964	Modbus_Comm_Error	MCER	Boolean	BOOL		False

2

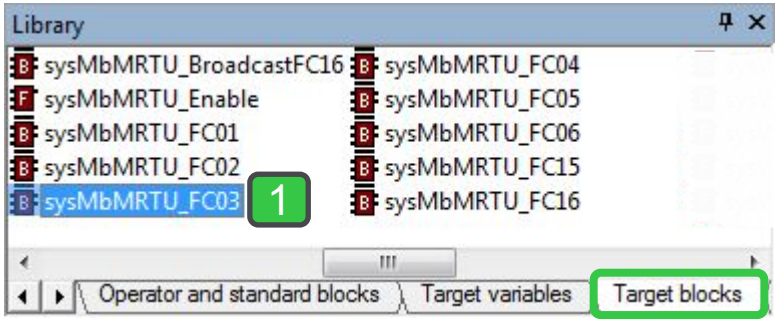
Write Multiple Registers, FC16

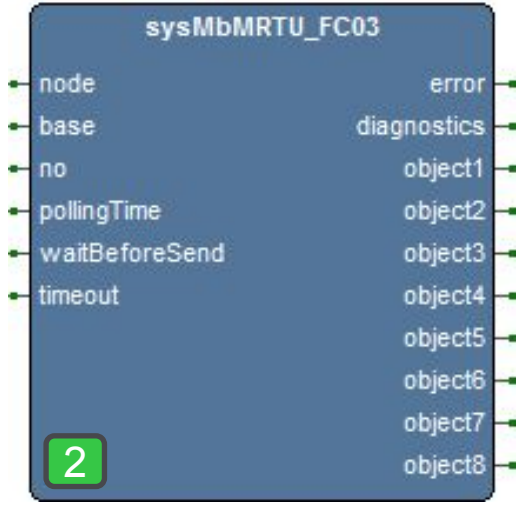
Master side, Assignment



Read Holding Registers, FC03

Master side, Input Pin-Out





Input:

Name	Type	Description
node	@MbMRTUNode	Slave node
base	UINT	Address of the first Holding Register to be read
no	UINT	Number of contiguous Holding Registers to be read
pollingTime	UINT	Polling time [ms]
waitBeforeSend	UINT	Time to wait before sending the message [ms]
timeout	UINT	Timeout [ms]

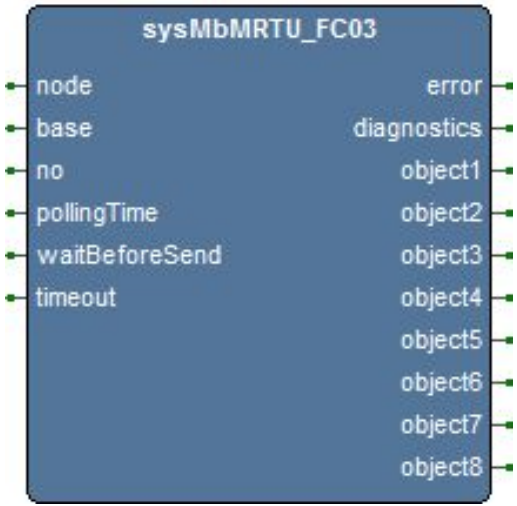
Note:

Read Holding Registers, FC03

Master side, Output Pin-Out

Output:

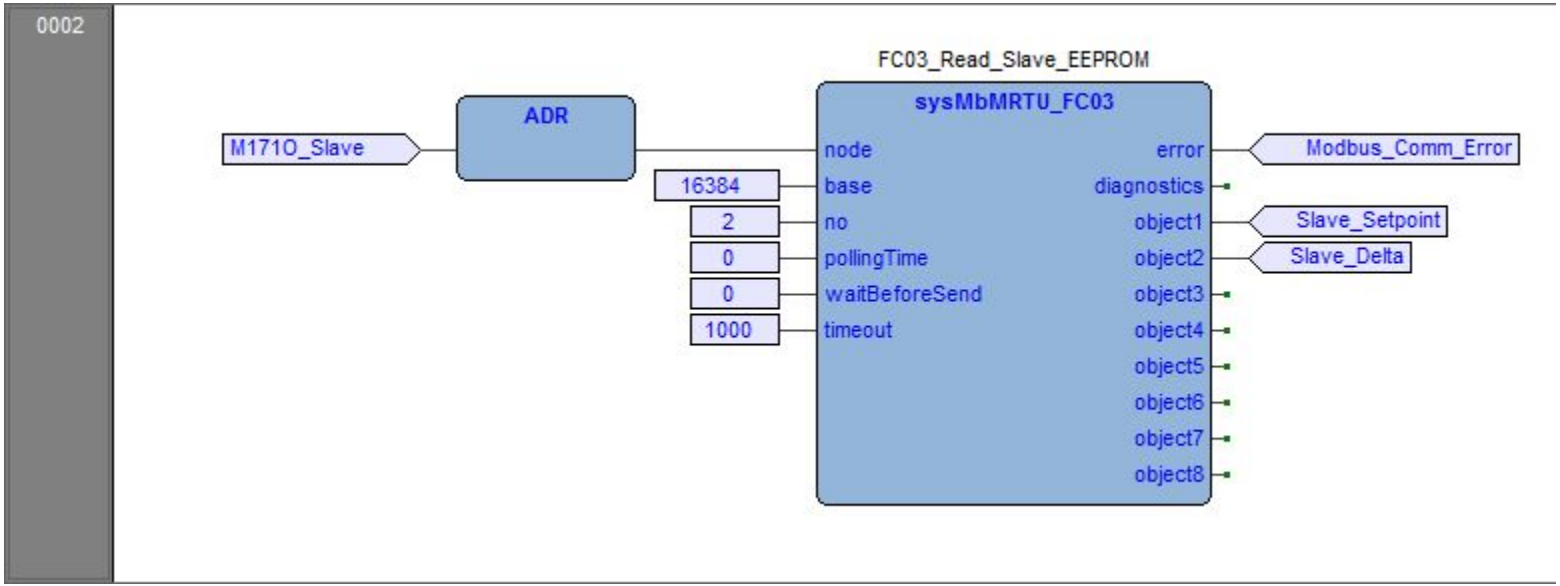
Name	Type	Description
error	BOOL	TRUE=error occurred,FALSE=no error
diagnostics	MbMRTUDiagnostics	Message diagnostics
object1	INT	1st Holding Register
object2	INT	2nd Holding Register
object3	INT	3rd Holding Register
object4	INT	4th Holding Register
object5	INT	5th Holding Register
object6	INT	6th Holding Register
object7	INT	7th Holding Register
object8	INT	8th Holding Register



Note:

Read Holding Registers, FC03

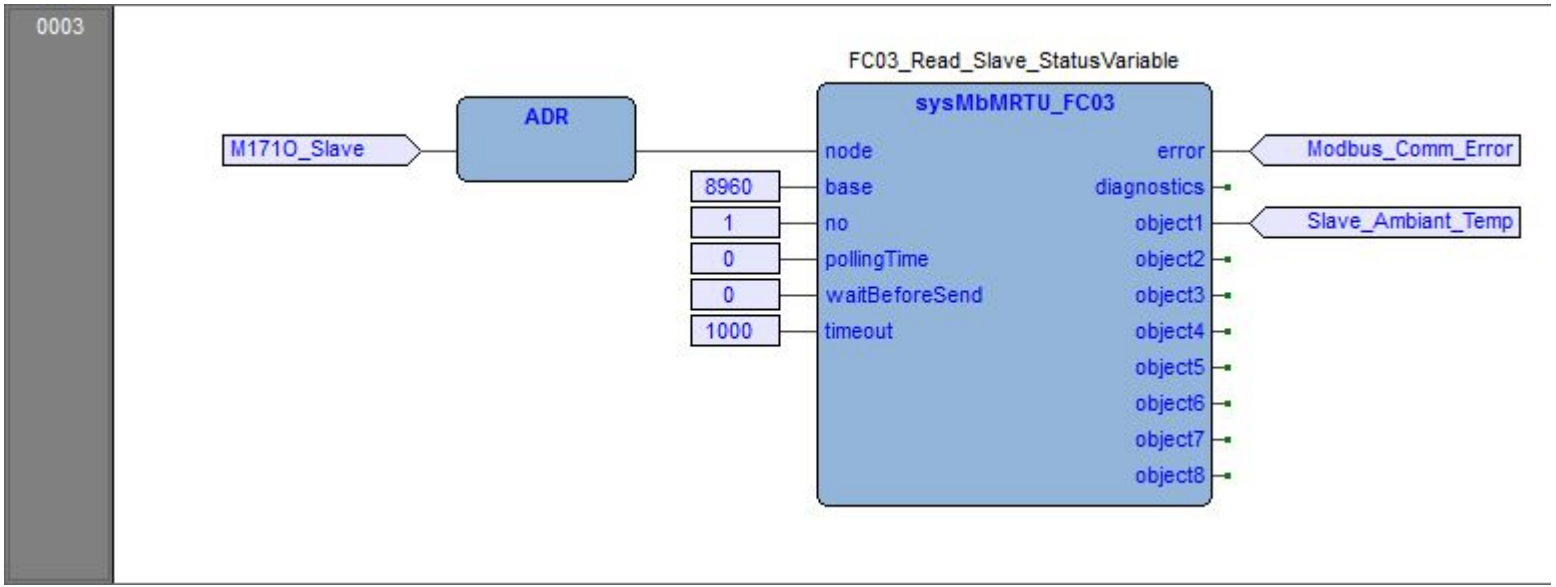
Master side, Assignment, Reading Slaves EEPROM Parameters



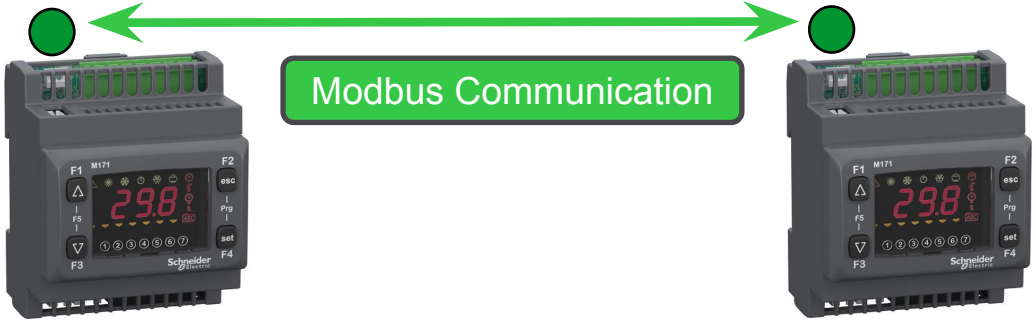
Note:

Read Holding Registers, FC03

Master side, Assignment, reading slave's ambient Temperature



Note:

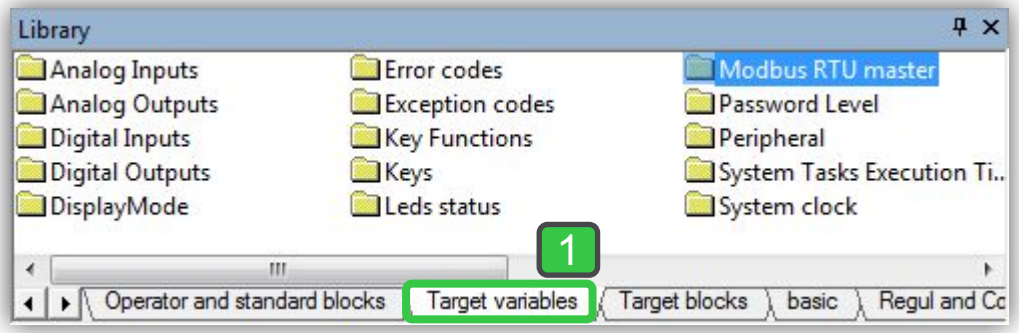


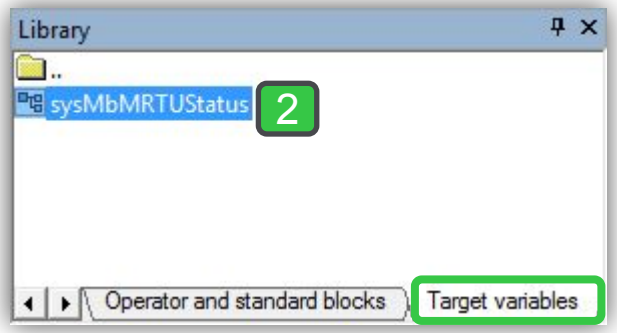
Modbus Communication Monitoring & Debugging

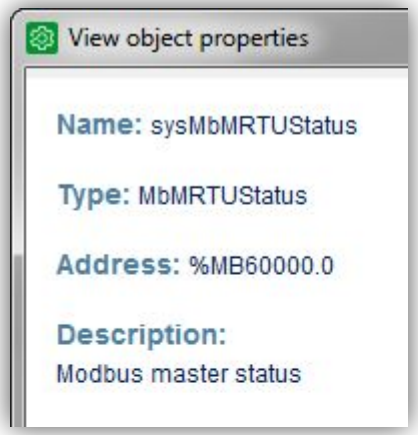
By Target Variables

Modbus Monitoring

Master Side, sysMbMRTUStatus



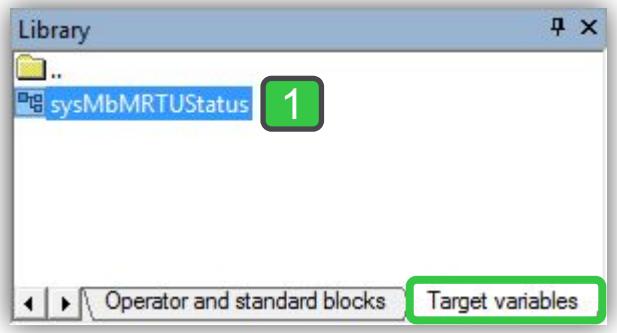




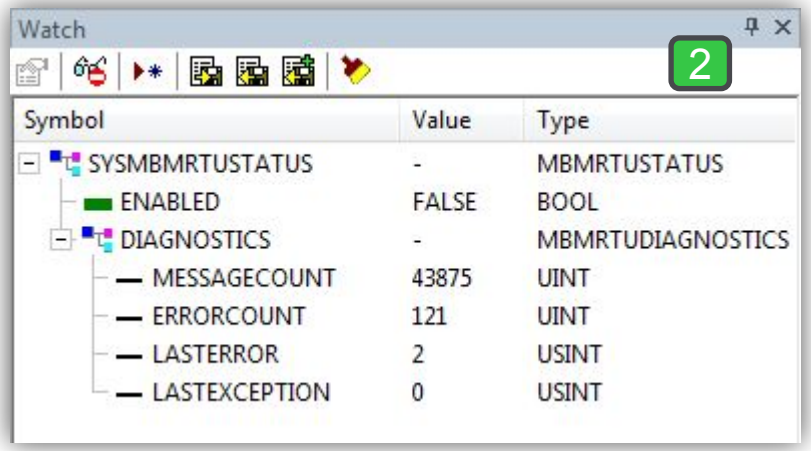
Note:

Modbus Communication Monitoring

Master side, Watch Window



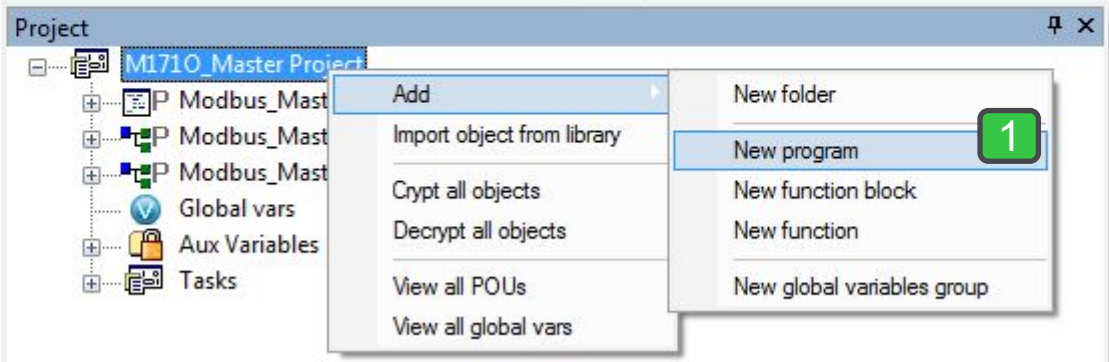
2. Drag & Drop

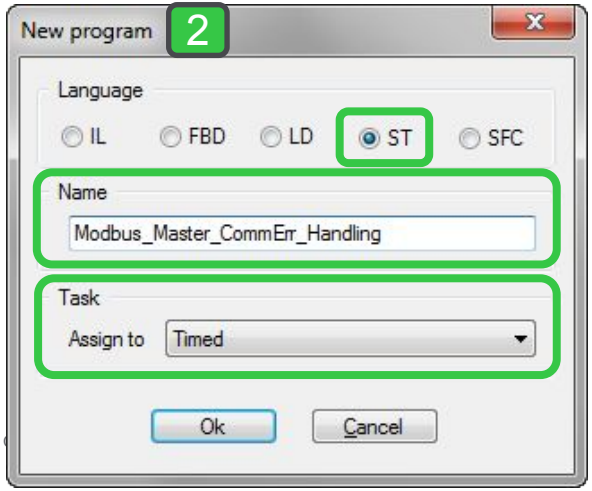


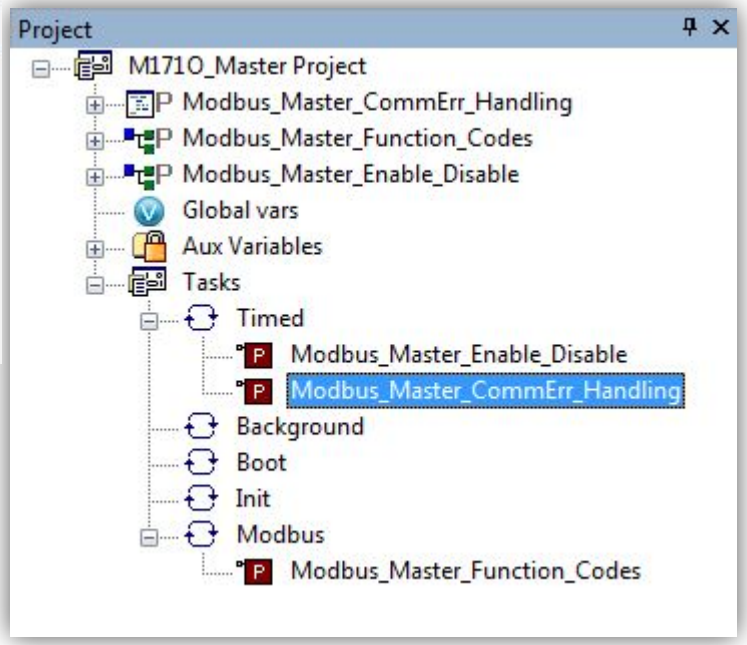
Note:

Modbus Master Communication Error handling/detection

Master Side, New Program







Note:

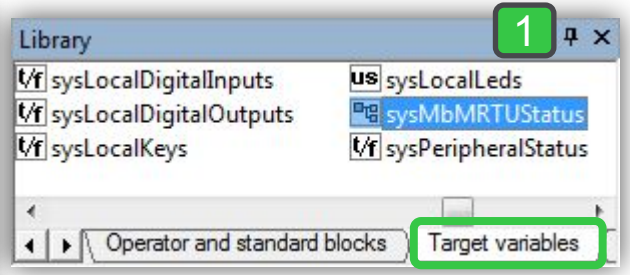
Monitoring Modbus Communication by third party

```
0001  (* Assigning structured target variable to the global
0002  status variable in order to monitor Modbus Master
0003  communication by third party *)
0004
0005  sysMbMRTUStatus.
0006
0007
0008
```

2

diagnostics
enabled

MbMRTUDiagnostics Master global diagnostics




```
0001  (* Assigning structured target variable to the global
0002  status variable in order to monitor Modbus Master
0003  communication by third party *)
0004
0005  sysMbMRTUStatus.diagnostics.
0006
0007
0008
0009
0010
```

3

errorCount
lastError
lastException
messageCount

UINT Message count

```
0001  (* Assigning structured target variable to the global
0002  status variable in order to monitor Modbus Master
0003  communication by third party *)
0004
0005  ErrorCount:= sysMbMRTUStatus.diagnostics.errorCount;
0006  MessageCount:= sysMbMRTUStatus.diagnostics.messageCount;
0007  LastError:= sysMbMRTUStatus.diagnostics.lastError;
0008  LastException:= sysMbMRTUStatus.diagnostics.lastException;
```

4

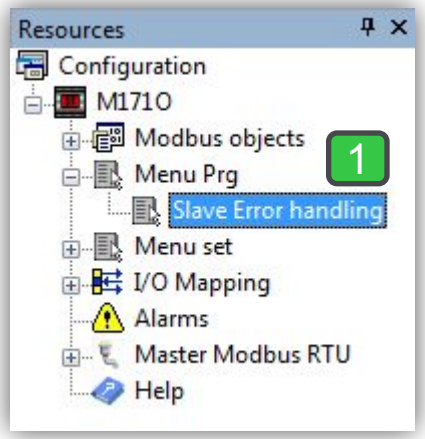
1. Drag & drop to from target variables list to the programming area

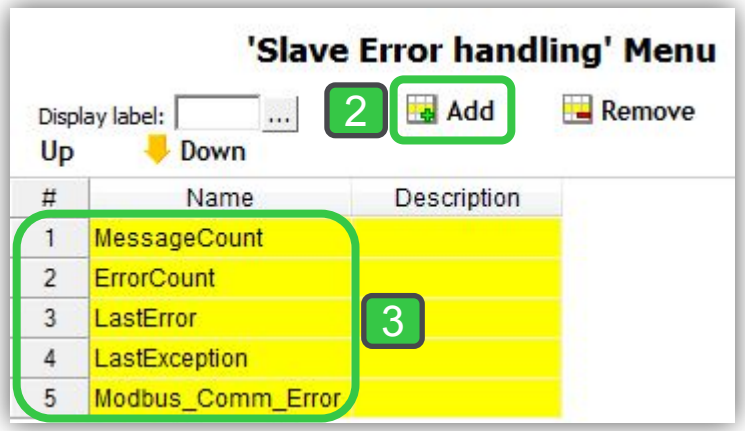
4. Assign desired global status variable to the diagnostic registers

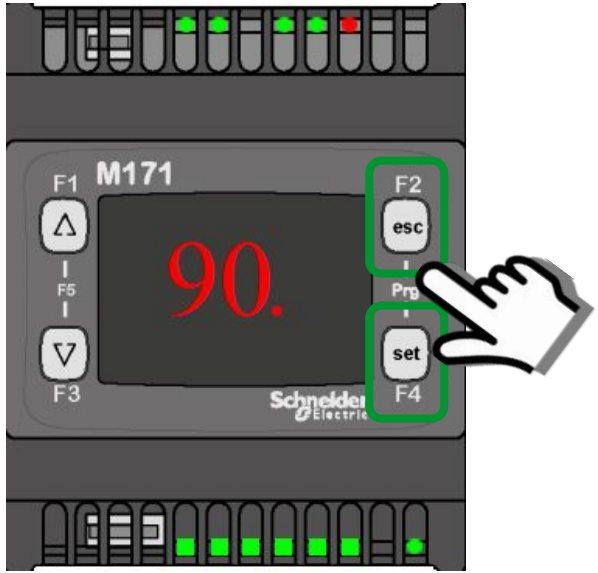
Life Is On Schneider Electric

Menu Prg.

Master Side, to access them via dashboard







5	8964	Modbus_Comm_Error	MCER
6	8965	MessageCount	SMGC
7	8966	ErrorCount	SERC
8	8967	LastError	SLER
9	8968	LastException	SLEX

4. Short Form/label

4

Note:

Appendix 1

Modbus Function codes Definition

Modbus Master Function Code

Read/write, **sysMbMRTU_FC01**

Read Coils - Function 01 (0x01)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num: 5

node : @MbMRTUNode (* Slave node *)

base : UINT (* Address of the first Coil to be read *)

no : UINT (* Number of contiguous Coils to be read *)

pollingTime : UINT (* Polling time [ms] *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 18

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

object1 : BOOL (* 1st Coil *)

...

object16 : BOOL (* 16th Coil *)

Extern vars num: 0

Modbus Master Function Code

Read/write, **sysMbMRTU_FC02**

Read Discrete Inputs - Function 02 (0x02)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num: 5

node : @MbMRTUNode (* Slave node *)

base : UINT (* Address of the first Discrete Input to be read *)

no : UINT (* Number of contiguous Discrete Inputs to be read *)

pollingTime : UINT (* Polling time [ms] *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 18

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

object1 : BOOL (* 1st Discrete Input *)

...

object16 : BOOL (* 16th Discrete Input *)

Extern vars num: 0

Modbus Master Function Code

Read/write, sysMbMRTU_FC03

Read Holding Registers - Function 03 (0x03)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num: 5

node : @MbMRTUNode (* Slave node *)

base : UINT (* Address of the first Holding Register to be read *)

no : UINT (* Number of contiguous Holding Registers to be read *)

pollingTime : UINT (* Polling time [ms] *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 10

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

object1 : INT (* 1st Holding Register *)

...

object8 : INT (* 8th Holding Register *)

Extern vars num: 0

Modbus Master Function Code

Read/write, sysMbMRTU_FC04

Read Input Registers - Function 04 (0x04)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num : 5

node : @MbMRTUNode (* Slave node *)

base : UINT (* Address of the first Input Register to be read *)

no : UINT (* Number of contiguous Input Registers to be read *)

pollingTime : UINT (* Polling time [ms] *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 10

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

object1 : INT (* 1st Input Register *)

...

object8 : INT (* 8th Input Register *)

Extern vars num: 0

Modbus Master Function Code

Read/write, **sysMbMRTU_FC05**

Write Single Coil - Function 05 (0x05)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num : 5

node : @MbMRTUNode (* Slave node *)

address : UINT (* Address of the Coil to write *)

object : BOOL (* Coil value *)

pollingTime : UINT (* Polling time [ms]. 0=write on variation *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 2

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

Extern vars num: 0

Modbus Master Function Code

Read/write, **sysMbMRTU_FC06**

Write Single Register - Function 06 (0x06)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num: 5

node : @MbMRTUNode (* Slave node *)

address : UINT (* Address of the Holding Register to write *)

object : INT (* Holding Register value *)

pollingTime : UINT (* Polling time [ms]. 0=write on variation *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 2

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

Extern vars num: 0

Modbus Master Function Code

Read/write, **sysMbMRTU_FC15**

Write Multiple Coils - Function 15 (0x0F)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num : 21

node : @MbMRTUNode (* Slave node *)

base : UINT (* Address of the first Coil to write *)

no : UINT (* Number of contiguous Coils to write *)

object1 : BOOL (* 1st Coil value *)

...

object16 : BOOL (* 16th Coil value *)

pollingTime : UINT (* Polling time [ms]. 0=write on variation *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 2

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

Extern vars num: 0

Modbus Master Function Code

Read/write, sysMbMRTU_FC16

Write Multiple Registers - Function 16 (0x10)

Type : FUNCTION BLOCK

Code type : EMBEDDED

Input vars num: 13

node : @MbMRTUNode (* Slave node *)

base : UINT (* Address of the first Holding Register to write *)

no : UINT (* Number of contiguous Holding Registers to write *)

object1 : INT (* 1st Holding Register value *)

...

object8 : INT (* 8th Holding Register value *)

pollingTime : UINT (* Polling time [ms]. 0=write on variation *)

timeout : UINT (* Timeout [ms] *)

Inout vars num: 0

Output vars num: 2

error : BOOL (* TRUE=error occurred,FALSE=no error *)

diagnostics : MbMRTUDiagnostics (* Message diagnostics *)

Extern vars num: 0

Questions?

Thank You