

VARIABLES

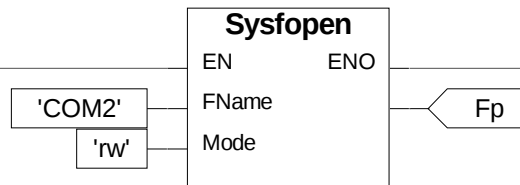
```
VAR_GLOBAL
TStatRd : ARRAY[ 0..1 ] OF TSTATDATARD; (* Dati letti da termostato *)
TStatWr : ARRAY[ 0..1 ] OF TSTATDATAWR; (* Dati scritti su termostato *)
END_VAR
```

	Project : HouseHeating	
	VARIABLES :	
	Release : FBModbusC	Ver :1.00
	Author :	Date:22/05/2012
	Note :	Page:1 of 1

```
VAR
Dummy : BOOL; (* Dummy variable *)
Fp : FILEP; (* File pointer *)
SMode : SetSMode; (* Set serial mode FB *)
TSTATCm : TSTATCommunication;
END_VAR
```

Eseguo set porta seriale da utilizzare

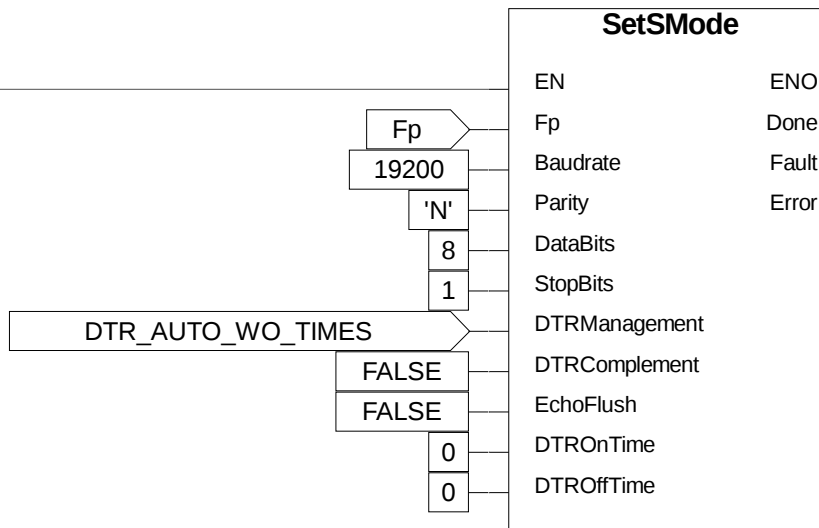
SysFirstLoop



Eseguo set parametri di comunicazione

SMode

SysFirstLoop



TSTATCm

TSTATCommunication

Enable Enabled

Fp
Read
Write

TStatRd

ADR

Fp

TStatWr

ADR

Project : HouseHeating

PROGRAM : Communication

Release : FBModbusC

Ver :1.00

Author :

Date:22/05/2012

Note :

Page:1 of 1

VAR_INPUT

```
Enable : BOOL; (* FB Enable *)
Fp : FILEP; (* File pointer *)
Read : @USINT; (* Buffer array valori letti *)
Write : @USINT; (* Buffer array valori scritti *)
END_VAR
```

VAR_OUTPUT

```
Enabled : BOOL; (* FB Enabled *)
END_VAR
```

VAR

```
CaseNr : USINT; (* Case number *)
IDx : USINT; (* Device index *)
Mdb : ModbusRTUMaster_v1; (* Modbus master FB *)
Node : ARRAY[ 0..15 ] OF USINT; (* Modbus nodes *)
Pulse : BOOL; (* Time base pulse *)
RdBuffer : TSTATDATARD; (* Buffer appoggio lettura *)
WrBuffer : TSTATDATAWR; (* Buffer appoggio scrittura *)
i : USINT; (* Auxiliary counter *)
SPtr : @USINT; (* Source pointer *)
DPtr : @USINT; (* Destination pointer *)
END_VAR
```

```
1 (* ***** *)
2 (* FUNCTION BLOCK "TSTATCommunication" *)
3 (* ***** *)
4 (* Questo blocco funzione gestisce la lettura da modbus master di un massimo *)
5 (* di 16 dispositivi TSTAT. *)
6 (* *)
7 (* Attenzione! Occorre definire gli indirizzi di nodo dei dispositivi. *)
8 (* Il dispositivo con indirizzo di nodo 16#FF termina la sequenza. *)
9 (* ----- *)
10 (* Il termostato TSTAT *)
11 (* Per entrare nella impostazione parametri da tastiera si tengono premuti *)
12 (* contemporaneamente i tasti OFF ed ON. *)
13 (* Nel menù parametri i tasti OFF ed ON permettono di selezionare parametro e *)
14 (* i tasi di setpoint permettono di definirne il valore. *)
15 (* *)
16 (* Di default il termostato comunica a 19200, n, 8, 1. *)
17 (* ----- *)
18
19 (* ----- *)
20 (* GESTIONE ABILITAZIONE *)
21 (* ----- *)
22 (* Eseguo gestione abilitazione. *)
23
24 IF NOT(Enable) THEN Enabled:=FALSE; RETURN; END_IF;
25
26 (* Eseguo inizializzazione al primo loop di abilitazione. *)
27
28 IF NOT(Enabled) THEN
29     Enabled:=TRUE; (* FB Enabled *)
30     IDx:=0; (* Device index *)
31     CaseNr:=0; (* Case number *)
32
33     (* Parametrizzo FB gestione modbus. *)
```

Project : HouseHeating

FUNCTION BLOCK : TSTATCommunication

Release : FBModbusC

Ver :1.00

Author :

Date:22/05/2012

Note :

Page:1 of 4

```

34
35     Mdb.File:=Fp; (* File pointer *)
36     Mdb.Timeout:=500; (* Tempo esecuzione comando (mS) *)
37     Mdb.Delay:=100; (* Tempo ritardo comando (mS) *)
38
39     (* Definisco nodi modbus. E' possibile in alternativa definirli *)
40     (* nel campo Init value di dichiarazione variabili. *)
41
42     Node[0]:=16#03; (* Modbus nodes *)
43     Node[1]:=16#FF; (* Modbus nodes *)
44 END_IF;
45
46 (* Eseguo FB modbus master. *)
47
48 Mdb(); (* Modbus master FB *)
49
50 (* ----- *)
51 (* CASES GESTIONE *)
52 (* ----- *)
53 (* Gestione lettura/scrittura dati su termostato. I dati in sono passati *)
54 (* alla FB in array di strutture dati una per i dati letti ed uno per i *)
55 (* dati da scrivere. Ogni struttura dati contiene 16 parametri ed ha una *)
56 (* dimensione 32 bytes. *)
57
58 CASE (CaseNr) OF
59
60     (* ----- *)
61     (* INIZIO COMUNICAZIONE *)
62     (* ----- *)
63     (* Trasferisco dati da buffer utente a buffer di appoggio. *)
64
65     0:
66     SPtr:=Write+(IDx*32); (* Source pointer *)
67     DPtr:=ADR(WrBuffer); (* Destination pointer *)
68     FOR i:=0 TO 31 DO
69         @DPtr:=@SPtr; (* Copia dati *)
70         SPtr:=SPtr+1; (* Source pointer *)
71         DPtr:=DPtr+1; (* Destination pointer *)
72     END_FOR;
73
74     CaseNr:=CaseNr+1; (* Case number *)
75
76     (* ----- *)
77     (* LETTURA ACQUISIZIONE DATI (Leggo 8 registri contigui) *)
78     (* ----- *)
79     (* Parametrizzo FB gestione lettura dati dai registri termostato. *)
80     (* Viene utilizzato il comando 03: Read holding registers. *)
81
82     1:
83     Mdb.Node:=Node[IDx]; (* Numero nodo modbus *)
84     Mdb.FCode:=16#03; (* Comando modbus *)
85     Mdb.Address:=101+1; (* Indirizzo variabili *)
86     Mdb.Points:=8; (* Numero di variabili *)
87     Mdb.Buffer:=ADR(RdBuffer.TRoom); (* Indirizzo buffer dati *)
88     Mdb.Enable:=TRUE; (* Abilitazione blocco funzione *)
89     CaseNr:=CaseNr+1; (* Case number *)
90
91     (* ----- *)
92     (* Attendo fine esecuzione comando modbus, copio dati letti in buffer *)
93     (* di ritorno. *)

```

	Project : HouseHeating	
	FUNCTION BLOCK : TSTATCommunication	
	Release : FBModbusC	Ver :1.00
	Author :	Date:22/05/2012
	Note :	Page:2 of 4

```

94
95 2:
96 IF NOT(Mdb.Done) THEN RETURN; END_IF;
97 Mdb.Enable:=FALSE; (* Abilitazione blocco funzione *)
98 CaseNr:=CaseNr+1; (* Case number *)
99
100 (* ----- *)
101 (* LETTURA ACQUISIZIONE SETPOINT (Leggo 1 registro) *)
102 (* ----- *)
103 (* Parametrizzo FB gestione lettura dati dai registri termostato. *)
104 (* Viene utilizzato il comando 03: Read holding registers. *)
105
106 3:
107 Mdb.Node:=Node[IDx]; (* Numero nodo modbus *)
108 Mdb.FCode:=16#03; (* Comando modbus *)
109 Mdb.Address:=135+1; (* Indirizzo variabili *)
110 Mdb.Points:=1; (* Numero di variabili *)
111 Mdb.Buffer:=ADR(RdBuffer.SetPoint); (* Indirizzo buffer dati *)
112 Mdb.Enable:=TRUE; (* Abilitazione blocco funzione *)
113 CaseNr:=CaseNr+1; (* Case number *)
114
115 (* ----- *)
116 (* Attendo fine esecuzione comando modbus, copio dati letti in buffer *)
117 (* di ritorno. *)
118
119 4:
120 IF NOT(Mdb.Done) THEN RETURN; END_IF;
121 Mdb.Enable:=FALSE; (* Abilitazione blocco funzione *)
122 CaseNr:=CaseNr+1; (* Case number *)
123
124 (* ----- *)
125 (* IMPOSTAZIONE SETPOINT *)
126 (* ----- *)
127 (* Per evitare di impostare valore di set point definito localmente *)
128 (* sul termostato l'impostazione viene eseguita solo se "SPWrite" è *)
129 (* attivo. Se comando attivo azzero flag di comando. *)
130
131 5:
132 IF (WrBuffer.SPWrite = 0) THEN CaseNr:=100; RETURN; END_IF;
133 DPtr:=Write+(IDx*32); (* Destination pointer *)
134 @DPtr:=0;
135
136 (* Parametrizzo FB gestione scrittura setpoint su termostato. *)
137 (* Viene utilizzato il comando 06: Preset single register. *)
138
139 WrBuffer.SPWrite:=0; (* Comando scrittura set point *)
140 Mdb.Node:=Node[IDx]; (* Numero di nodo modbus *)
141 Mdb.FCode:=16#06; (* Comando modbus *)
142 Mdb.Address:=135+1; (* Indirizzo variabili *)
143 Mdb.Points:=1; (* Numero di variabili *)
144 Mdb.Buffer:=ADR(WrBuffer.SetPoint); (* Indirizzo buffer dati *)
145 Mdb.Enable:=TRUE; (* Abilitazione blocco funzione *)
146 CaseNr:=CaseNr+1; (* Case number *)
147
148 (* ----- *)
149 (* Attendo fine esecuzione comando modbus. *)
150
151 6:
152 IF NOT(Mdb.Done) THEN RETURN; END_IF;
153 Mdb.Enable:=FALSE; (* Abilitazione blocco funzione *)

```

Project : HouseHeating

FUNCTION BLOCK : TSTATCommunication

Release : FBModbusC

Ver :1.00

Author :

Date:22/05/2012

Note :

Page:3 of 4

FUNCTION_BLOCK TSTATCommunication

```

154 CaseNr:=100; (* Case number *)
155
156 (* ----- *)
157 (* FINE COMUNICAZIONE *)
158 (* ----- *)
159 (* Fine comunicazione trasferisco dati da buffer appoggio in buffer *)
160 (* utente passato alla FB. *)
161
162 100:
163 SPtr:=ADR(RdBuffer); (* Source pointer *)
164 DPtr:=Read+(IDx*32); (* Destination pointer *)
165 FOR i:=0 TO 31 DO
166     @DPtr:=@SPtr; (* Copia dati *)
167     SPtr:=SPtr+1; (* Source pointer *)
168     DPtr:=DPtr+1; (* Destination pointer *)
169 END_FOR;
170
171 (* Eseguo incremento pointer. *)
172
173 IDx:=IDx+1; (* Device index *)
174 IF (IDx >= 16) THEN IDx:=0; END_IF;
175 IF (Node[IDx] = 16#FF) THEN IDx:=0; END_IF;
176 CaseNr:=0; (* Case number *)
177 END_CASE;
178
179 (* [End of file] *)
180
181

```

	Project : HouseHeating	
	FUNCTION BLOCK : TSTATCommunication	
	Release : FBModbusC	Ver :1.00
	Author :	Date:22/05/2012
	Note :	Page:4 of 4

(SFR054B000) Set the serial communication mode
ENCRYPTED CODE

```
VAR_INPUT
Fp : FILEP;
Baudrate : UDINT; (* Baudrate *)
Parity : STRING[ 1 ]; (* Parity type *)
DataBits : USINT; (* Nr of data bits *)
StopBits : USINT; (* Nr of stop bits *)
DTRManagement : USINT; (* DTR management type *)
DTRComplement : BOOL; (* Complement the DTR signal *)
EchoFlush : BOOL; (* Flush the echo *)
DTROnTime : UINT; (* DTR On wait time *)
DTROffTime : UINT; (* DTR Off wait time *)
END_VAR

VAR_OUTPUT
Done : BOOL; (* Execution done *)
Fault : BOOL; (* Execution fault *)
END_VAR
```

1

	Project : HouseHeating	
	FUNCTION BLOCK : SetSMode	
	Release : FBModbusC	Ver :1.00
	Author :	Date:22/05/2012
	Note :	Page:1 of 1

(SFR054B100) Manages the modbus RTU master communication
ENCRYPTED CODE

```
VAR_INPUT
Enable : BOOL;
File : FILEP; (* Terminal I/O pointer *)
Node : USINT; (* Node number *)
FCode : USINT; (* Function code *)
Address : UINT; (* Start address *)
Points : USINT; (* Number of points *)
Buffer : @USINT; (* Address of data buffer *)
Timeout : UINT; (* Timeout time (mS) *)
Delay : UINT; (* Delay time (mS) *)
END_VAR
```

```
VAR_OUTPUT
Done : BOOL; (* Command done *)
Ok : BOOL := FALSE; (* Execution Ok *)
Fault : BOOL; (* Command fault *)
Errors : UDINT; (* Error counter *)
END_VAR
```

Project : HouseHeating

FUNCTION BLOCK : ModbusRTUMaster_v1

Release : FBModbusC

Ver :1.00

Author :

Date:22/05/2012

Note :

Page:1 of 1