

VARIABLES

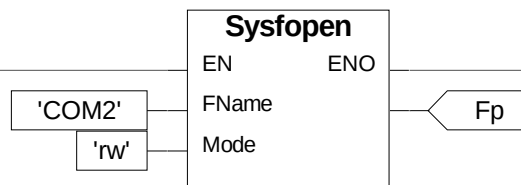
```
VAR_GLOBAL
Inputs : ARRAY[ 0..15 ] OF IOBANK;
Outputs : ARRAY[ 0..15 ] OF IOBANK;
END_VAR
```

	Project : FBModbusCm	
	VARIABLES :	
	Release : FBModbusC	Ver :1.00
	Author :	Date:20/04/2012
	Note :	Page:1 of 1

```
VAR
Dummy : BOOL; (* Dummy variable *)
Fp : FILEP; (* File pointer *)
MdbCm : FBModbusCm; (* FB comunicazione modbus *)
SMode : SetSMode; (* Set serial mode FB *)
END_VAR
```

Eseguo set porta seriale da utilizzare

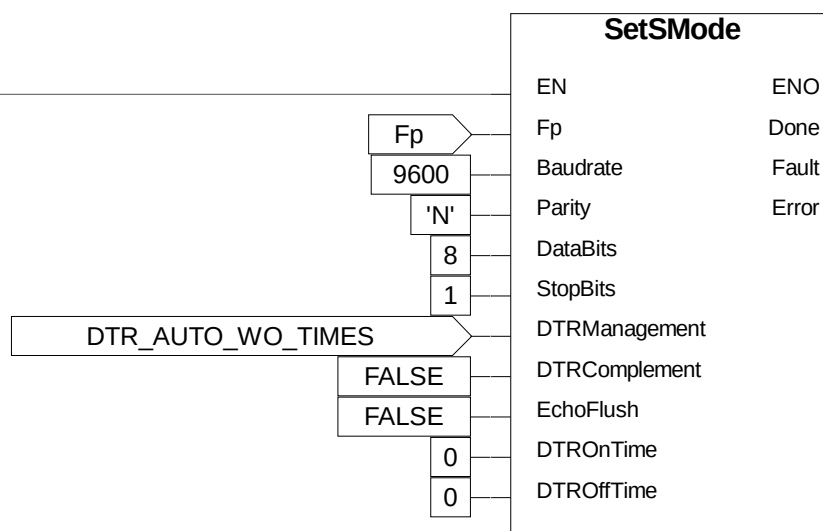
SysFirstLoop



Eseguo set parametri di comunicazione

SMode

SysFirstLoop



MdbCm

FBModbusCm

Enable Enabled

Fp
Inputs
Outputs

Inputs

ADR

Fp

Outputs

ADR

Project : FBModbusCm

PROGRAM : Communication

Release : FBModbusC

Ver :1.00

Author :

Date:20/04/2012

Note :

Page:1 of 1

```
1
2
3 Outputs[0].IO0:=NOT(Inputs[0].IO0);
4 Outputs[0].IO1:=NOT(Inputs[0].IO1);
5 Outputs[0].IO2:=NOT(Inputs[0].IO2);
6 Outputs[0].IO3:=NOT(Inputs[0].IO3);
7
8 Outputs[0].IO0:=SysClock1000;
9
10
```

	Project : FBModbusCm	
	PROGRAM : Logic	
	Release : FBModbusC	Ver :1.00
	Author :	Date:20/04/2012
	Note :	Page:1 of 1

VAR_INPUT

```

Enable : BOOL; (* FB Enable *)
Fp : FILEP; (* File pointer *)
Inputs : @UDINT; (* Buffer array ingressi digitali *)
Outputs : @UDINT; (* Buffer array uscite digitali *)
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 *)
END_VAR

```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "ModbusSlavesCm" *)
3 (* ***** *)
4 (* Questo blocco funzione gestisce la lettura da modbus master di un massimo *)
5 (* di 16 dispositivi. *)
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
11 (* ----- *)
12 (* GESTIONE ABILITAZIONE *)
13 (* ----- *)
14 (* Eseguo gestione abilitazione. *)
15
16 IF NOT(Enable) THEN Enabled:=FALSE; RETURN; END_IF;
17
18 (* Eseguo inizializzazione al primo loop di abilitazione. *)
19
20 IF NOT(Enabled) THEN
21     Enabled:=TRUE; (* FB Enabled *)
22     IDx:=0; (* Device index *)
23     CaseNr:=0; (* Case number *)
24
25     (* Parametrizzo FB gestione modbus. *)
26
27     Mdb.File:=Fp; (* File pointer *)
28     Mdb.Timeout:=500; (* Tempo esecuzione comando (mS) *)
29     Mdb.Delay:=10; (* Tempo ritardo comando (mS) *)
30
31     (* Definisco nodi modbus. E' possibile in alternativa definirli *)
32     (* nel campo Init value di dichiarazione variabili. *)
33
34     Node[0]:=16#01; (* Modbus nodes *)
35     Node[1]:=16#05; (* Modbus nodes *)
36     Node[2]:=16#FF; (* Modbus nodes *)
37 END_IF;
38

```

Project : FBModbusCm

FUNCTION BLOCK : FBModbusCm

Release : FBModbusC

Ver :1.00

Author :

Date:20/04/2012

Note :

Page:1 of 3

```

39  (* Eseguo FB modbus master. *)
40
41  Mdb(); (* Modbus master FB *)
42
43  (* ----- *)
44  (* CASES GESTIONE *)
45  (* ----- *)
46  (* Eseguo acquisizione comandi accensione dai files CSV. *)
47
48  CASE (CaseNr) OF
49
50      (* ----- *)
51      (* Parametrizzo FB gestione modbus per lettura ingressi. *)
52      (* Viene utilizzato il comando 02: Read input status. *)
53
54      0:
55      Mdb.Node:=Node[IDx]; (* Numero nodo modbus *)
56      Mdb.FCode:=16#02; (* Comando modbus *)
57      Mdb.Address:=16#00+1; (* Indirizzo variabili *)
58      Mdb.Points:=4; (* Numero di variabili *)
59      Mdb.Buffer:=Inputs+(IDx*4); (* Indirizzo buffer dati *)
60      Mdb.Enable:=TRUE; (* Abilitazione blocco funzione *)
61      CaseNr:=CaseNr+1; (* Case number *)
62
63      (* ----- *)
64      (* Attendo fine esecuzione comando modbus. *)
65
66      1:
67      IF NOT(Mdb.Done) THEN RETURN; END_IF;
68      Mdb.Enable:=FALSE; (* Abilitazione blocco funzione *)
69      CaseNr:=CaseNr+1; (* Case number *)
70
71      (* ----- *)
72      (* Parametrizzo FB gestione modbus per scrittura uscite. *)
73      (* Viene utilizzato il comando 0F: Force Multiple Coils. *)
74
75      2:
76      Mdb.Node:=Node[IDx]; (* Numero di nodo modbus *)
77      Mdb.FCode:=16#0F; (* Comando modbus *)
78      Mdb.Address:=0+1; (* Indirizzo variabili *)
79      Mdb.Points:=4; (* Numero di variabili *)
80      Mdb.Buffer:=Outputs+(IDx*4); (* Indirizzo buffer dati *)
81      Mdb.Enable:=TRUE; (* Abilitazione blocco funzione *)
82      CaseNr:=CaseNr+1; (* Case number *)
83
84      (* ----- *)
85      (* Attendo fine esecuzione comando modbus. *)
86
87      3:
88      IF NOT(Mdb.Done) THEN RETURN; END_IF;
89      Mdb.Enable:=FALSE; (* Abilitazione blocco funzione *)
90      CaseNr:=CaseNr+1; (* Case number *)
91
92      (* ----- *)
93      (* Incremento indice strumento da gestire e controllo se terminato. *)
94
95      4:
96      IDx:=IDx+1; (* Device index *)
97      IF (IDx >= 16) THEN IDx:=0; END_IF;
98      IF (Node[IDx] = 16#FF) THEN IDx:=0; END_IF;

```

Project : FBMModbusCm

FUNCTION BLOCK : FBMModbusCm

Release : FBMModbusC

Ver :1.00

Author :

Date:20/04/2012

Note :

Page:2 of 3

FUNCTION_BLOCK FBModbusCm

```

99      CaseNr:=0; (* Case number *)
100      END_CASE;
101
102 (* [End of file] *)
103
104

```

	Project : FBModbusCm	
	FUNCTION BLOCK : FBModbusCm	
	Release : FBModbusC	Ver :1.00
	Author :	Date:20/04/2012
	Note :	Page:3 of 3

(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 : FBModbusCm

FUNCTION BLOCK : SetSMode

Release : FBModbusC

Ver :1.00

Author :

Date:20/04/2012

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 : FBModbusCm

FUNCTION BLOCK : ModbusRTUMaster_v1

Release : FBModbusC

Ver :1.00

Author :

Date:20/04/2012

Note :

Page:1 of 1