

VARIABLES

```
VAR_GLOBAL
Quote : UINT; (* Quota encoder (Tacche) *)
END_VAR
```

	Project : EncoderToModbus	
	VARIABLES :	
	Release : EncoderToM	Ver :1.00
	Author :	Date:11/03/2014
	Note :	Page:1 of 1

```
VAR
DInputs : SysGetPhrDI; (* Digital inputs acquisistion *)
EQuote : IOEncoder; (* Encoder quote *)
END_VAR
```

```
1 (* ***** *)
2 (* ACQUISIZIONE ENCODER *)
3 (* ***** *)
4 (* Programma è eseguito in task Fast per poter gestire encoder fino a 400 Hz. *)
5 (* ----- *)
6
7 (* ----- *)
8 (* INIZIALIZZAZIONI *)
9 (* ----- *)
10 (* Eseguo inizializzazioni. *)
11
12 IF (SysFirstLoop) THEN
13     DInputs.Address:=255; (* Module address *)
14     DInputs.Mode:=DI_I_8_LL; (* Acquisition mode *)
15 END_IF;
16
17 (* ----- *)
18 (* GESTIONE QUOTA ENCODER *)
19 (* ----- *)
20 (* Eseguo gestione quota encoder. *)
21
22 DInputs(); (* Eseguo FB lettura ingressi *)
23
24 EQuote.APhase:=((DInputs.Value AND 16#01) <> 0); (* Input 0 fase "A" *)
25 EQuote.BPhase:=((DInputs.Value AND 16#02) <> 0); (* Input 1 fase "B" *)
26 EQuote.Enable:=TRUE; (* Eseguo gestione encoder *)
27 Quote:=TO_UINT(EQuote.Quote); (* Quota encoder (Tacche) *)
28
29 (* [End of file] *)
30
```

	Project : EncoderToModbus	
	PROGRAM : EncoderAcq	
	Release : EncoderToM	Ver :1.00
	Author :	Date:11/03/2014
	Note :	Page:1 of 1

VAR

```

Fp : FILEP; (* File pointer *)
ABf : BOOL; (* Auxiliary buffer *)
Sm : SYSSERIALMODE; (* Serial mode *)
MyModbus : ModbusMaster; (* Modbus master *)
END_VAR

```

```

1 (* ***** *)
2 (* COMUNICAZIONE MODBUS *)
3 (* ***** *)
4 (* Programma è eseguito in task Back per gestire comunicazioni seriali. *)
5 (* ----- *)
6
7 (* ----- *)
8 (* INIZIALIZZAZIONI *)
9 (* ----- *)
10 (* Eseguo inizializzazioni. *)
11
12 IF (SysFirstLoop) THEN
13     Fp:=Sysfopen('COM0', 'rw'); (* File pointer *)
14
15     (* Imposto parametri comunicazione. *)
16
17     ABf:=SysGetSerialMode(ADR(Sm), Fp);
18     Sm.Baudrate:=19200; (* Baud rate *)
19     Sm.DataBits:=8; (* Data bits *)
20     Sm.Parity:='E'; (* Parità *)
21     Sm.StopBits:=1; (* Stop bits *)
22     Sm.DTRManagement:=DTR_AUTO_WO_TIMES; (* DTR management mode *)
23     ABf:=SysSetSerialMode(ADR(Sm), Fp);
24
25     (* Inizializzo FB modbus. *)
26
27     MyModbus.File:=Fp; (* File pointer *)
28     MyModbus.Type:=0; (* Modbus type RTU *)
29     MyModbus.IFTime:=1720; (* Interframe time *)
30     MyModbus.Timeout:=500; (* Timeout time *)
31     MyModbus.Delay:=1000; (* Pause time *)
32
33     MyModbus.Node:=16#01; (* Node address *)
34     MyModbus.FCode:=16#10; (* Function code *)
35     MyModbus.Address:=16#A302; (* Register address *)
36
37     MyModbus.Buffer:=ADR(Quote); (* Buffer address *)
38     MyModbus.Points:=1; (* Number of points *)
39 END_IF;
40
41 (* ----- *)
42 (* GESTIONE COMUNICAZIONE MODBUS *)
43 (* ----- *)
44 (* Eseguo comunicazione modbus. *)
45
46 MyModbus(Enable:=NOT(MyModbus.Done)); (* Eseguo FB modbus *)
47
48 (* [End of file] *)
49
50

```

Project : EncoderToModbus

PROGRAM : ModbusCmm

Release : EncoderToM

Ver :1.00

Author :

Date:11/03/2014

Note :

Page:1 of 1

FUNCTION_BLOCK IOEncoder

(SFR054B220) Acquires a quadrature encoder over logic inputs
ENCRYPTED CODE

```
VAR_INPUT
Enable : BOOL; (* FB enable *)
Reset : BOOL; (* Quote reset *)
APhase : BOOL; (* Encoder A phase *)
BPhase : BOOL; (* Encoder B phase *)
END_VAR

VAR_OUTPUT
Enabled : BOOL; (* FB enabled *)
Error : BOOL; (* Encoder acquisition error *)
Quote : UDINT; (* Encoder quote *)
END_VAR
```

1

	Project : EncoderToModbus	
	FUNCTION BLOCK : IOEncoder	
	Release : EncoderToM	Ver :1.00
	Author :	Date:11/03/2014
	Note :	Page:1 of 1

(SFR054B220) Manages the modbus master communication
ENCRYPTED CODE

```
VAR_INPUT
Enable : BOOL;
SpyOn : BOOL; (* Spy active *)
File : FILEP; (* Terminal I/O pointer *)
Type : USINT; (* Modbus type *)
Node : USINT; (* Node number *)
FCode : USINT; (* Function code *)
Address : UINT; (* Start address *)
Points : UDINT; (* Number of points *)
Buffer : @USINT; (* Address of data buffer *)
IFTime : UDINT; (* Interframe time (uS) *)
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 : EncoderToModbus	
	FUNCTION BLOCK : ModbusMaster	
	Release : EncoderToM	Ver :1.00
	Author :	Date:11/03/2014
	Note :	Page:1 of 1