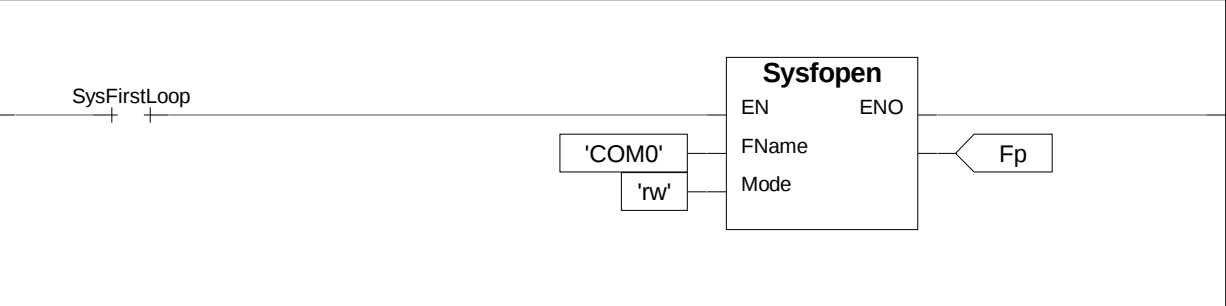
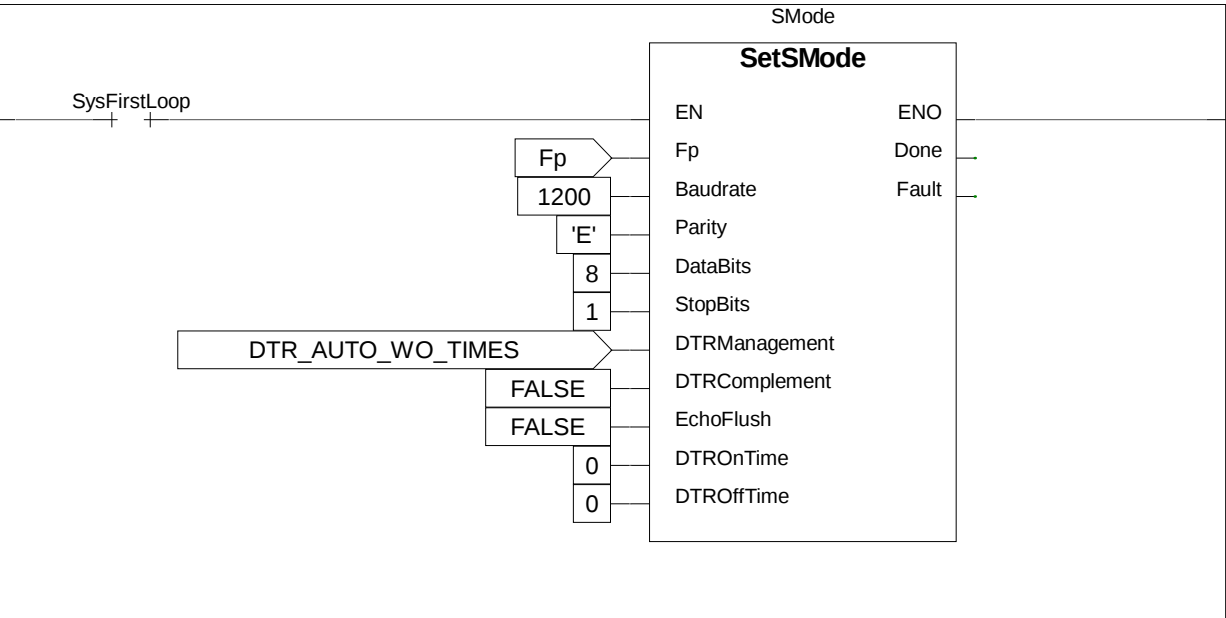


```
VAR
Fp : FILEP; (* File pointer *)
SMode : SetSMode; (* Set serial mode FB *)
Done : ARRAY[ 0..1 ] OF BOOL; (* Sensor acquisition done *)
Sensor1 : RHTSensor; (* RHTSensor FB *)
Sensor2 : RHTSensor; (* RHTSensor FB *)
Temperature : ARRAY[ 0..1 ] OF REAL; (* Valore temperatura (°C) *)
Humidity : ARRAY[ 0..1 ] OF REAL; (* Valore umidità (%) *)
DewPoint : ARRAY[ 0..1 ] OF REAL; (* Punto di rugiada (°C) *)
END_VAR
```

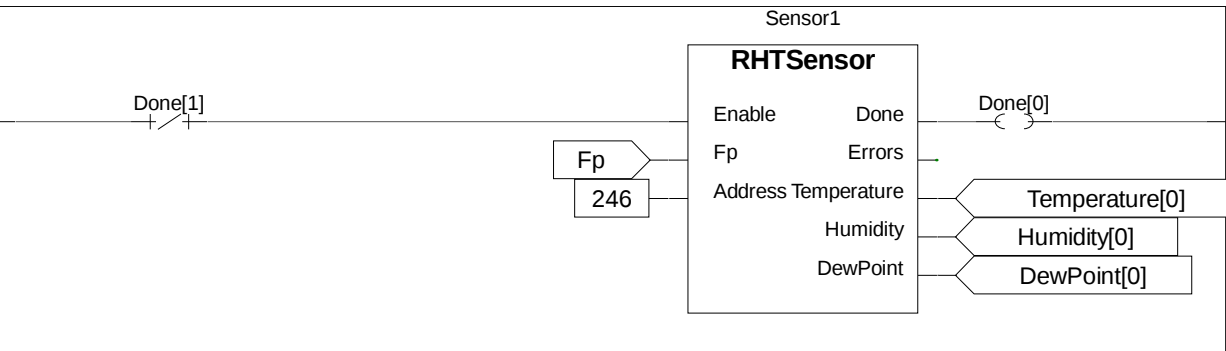
1



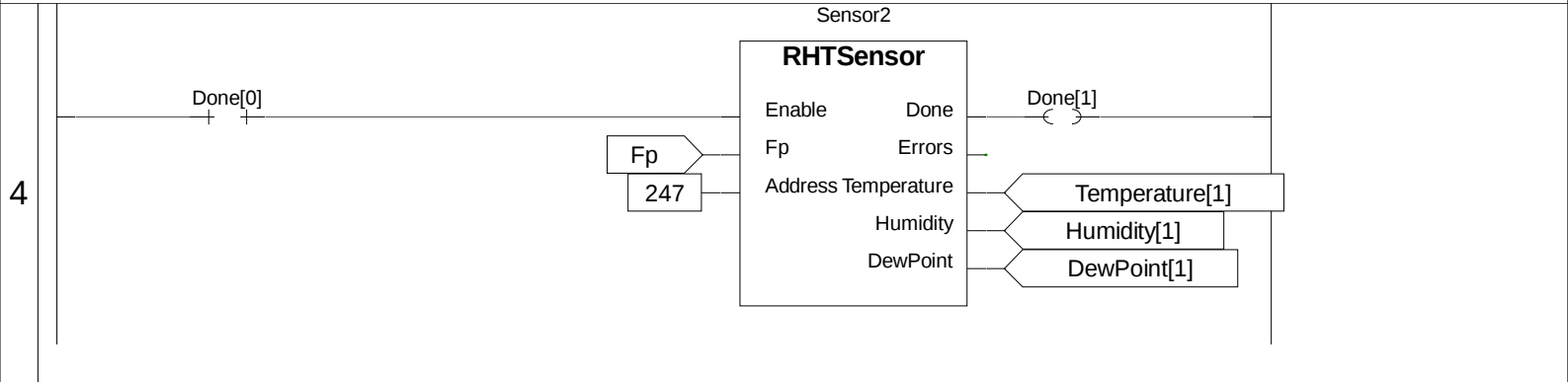
2



3



Project : RHSensor	
PROGRAM : RHTSensorRd	
Release : RHSensor	Ver :1.00
Author :	Date:21/02/2013
Note :	Page:1 of 2



	Project : RHTSensor	
	PROGRAM : RHTSensorRd	
	Release : RHTSensor	Ver :1.00
	Author :	Date:21/02/2013
	Note :	Page:2 of 2

(SFR054B200) 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 : RHSensor

FUNCTION BLOCK : ModbusRTUMaster_v1

Release : RHSensor

Ver :1.00

Author :

Date:21/02/2013

Note :

Page:1 of 1

(SFR054B200) 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 : RHSensor	
	FUNCTION BLOCK : SetSMode	
	Release : RHSensor	Ver :1.00
	Author :	Date:21/02/2013
	Note :	Page:1 of 1

VAR_INPUT

```
Enable : BOOL; (* FB Enable *)
Fp : FILEP; (* File pointer *)
Address : USINT; (* Address sensore *)
END_VAR
```

VAR_OUTPUT

```
Done : BOOL; (* FB Done *)
Errors : UDINT; (* Numero errori comunicazione *)
Temperature : REAL; (* Temperatura (°C) *)
Humidity : REAL; (* Umidità (%) *)
DewPoint : REAL; (* Punto di rugiada (°C) *)
END_VAR
```

VAR

```
Mdb : ModbusRTUMaster_v1; (* Modbus RTU Master *)
RxBuffer : ARRAY[ 0..3 ] OF UINT; (* Rx buffer *)
END_VAR
```

```
1 (* ***** *)
2 (* FUNCTION BLOCK "RHTSensor" *)
3 (* ***** *)
4 (* Questo blocco funzione esegue la lettura tramite protocollo modbus di un *)
5 (* sensore di umidità e temperatura RHT. *)
6 (* *)
7 (* ----- *)
8
9 (* ----- *)
10 (* ESEGUO CONFIGURAZIONE MODBUS *)
11 (* ----- *)
12 (* Eseguo configurazione modbus leggo solo i 3 registri necessari. *)
13 (* 07: Valore temperatura (Decimi di grado) *)
14 (* 08: Valore umidità (Decimi di RH%) *)
15 (* 09: Valore punto di rugiada (Decimi di grado) *)
16
17 Mdb.File:=Fp; (* File pointer *)
18 Mdb.Node:=Address; (* Modbus node *)
19 Mdb.FCode:=16#03; (* Function code *)
20 Mdb.Buffer:=ADR(RxBuffer); (* Buffer address *)
21 Mdb.Points:=3; (* Numero registri da leggere *)
22 Mdb.Address:=7+1; (* Address registro modbus *)
23 Mdb.Enable:=TRUE; (* Abilitazione modbus *)
24 Mdb.Timeout:=500; (* Timeout *)
25 Mdb.Delay:=1000; (* Timeout *)
26
27 (* ----- *)
28 (* ESEGUO GESTIONE MODBUS *)
29 (* ----- *)
30 (* Eseguo gestione modbus. *)
31
32 Mdb.Enable:=Enable; (* Enable *)
33 Mdb(); (* Gestione modbus master *)
34 Done:=Mdb.Done; (* FB Done *)
35 Errors:=Mdb.Errors; (* Numero errori comunicazione *)
36
37 (* Eseguo conversione valori acquisiti dal sensore. *)
38
```

Project : RHTSensor

FUNCTION_BLOCK : RHTSensor

Release : RHTSensor

Ver :1.00

Author :

Date:21/02/2013

Note :

Page:1 of 2

FUNCTION_BLOCK RHTSensor

```

39     IF (Mdb.Ok) THEN
40         Temperature:=TO_REAL(RxBuffer[0])/10.0; (* Temperatura (°C) *)
41         Humidity:=TO_REAL(RxBuffer[1])/10.0; (* Umidità (%) *)
42         DewPoint:=TO_REAL(RxBuffer[2])/10.0; (* Punto di rugiada (°C) *)
43     END_IF;
44
45 (* [End of file] *)
46

```

	Project : RHTSensor	
	FUNCTION_BLOCK : RHTSensor	
	Release : RHTSensor	Ver :1.00
	Author :	Date:21/02/2013
	Note :	Page:2 of 2