

VARIABLES

VAR_GLOBAL

TS0_ROMID AT %MB100.0 : ARRAY[0..7] OF BYTE; (* ID Sensore 0 *)

TS1_ROMID AT %MB100.8 : ARRAY[0..7] OF BYTE; (* ID Sensore 1 *)

TS2_ROMID AT %MB100.16 : ARRAY[0..7] OF BYTE; (* ID Sensore 2 *)

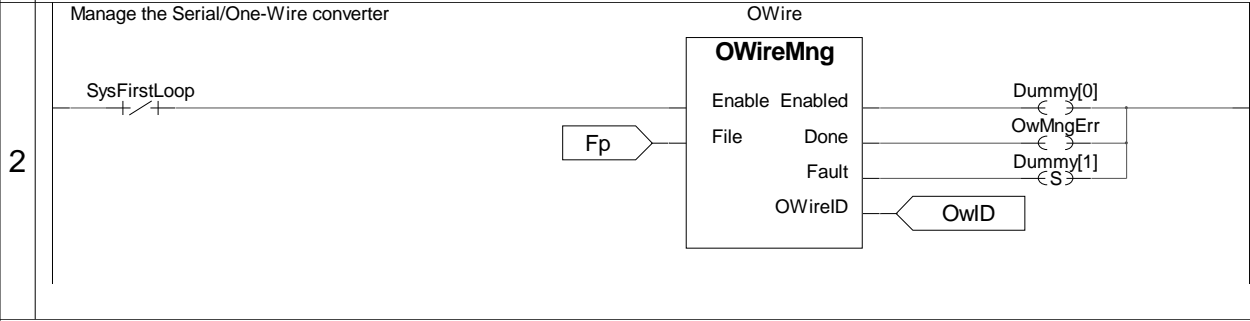
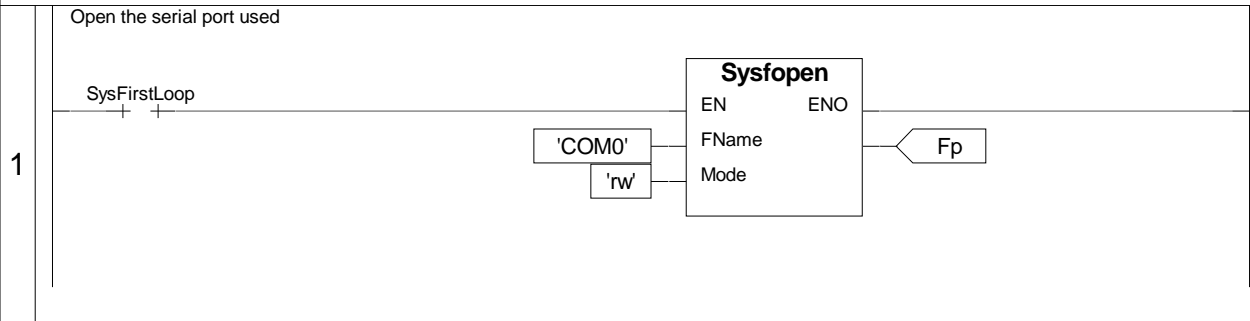
Sensor_Temp_Array : ARRAY[0..2] OF REAL;

END_VAR

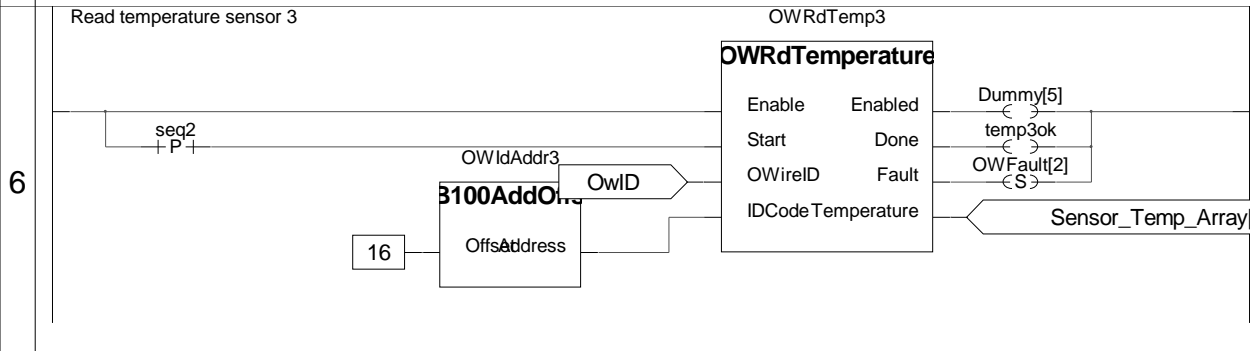
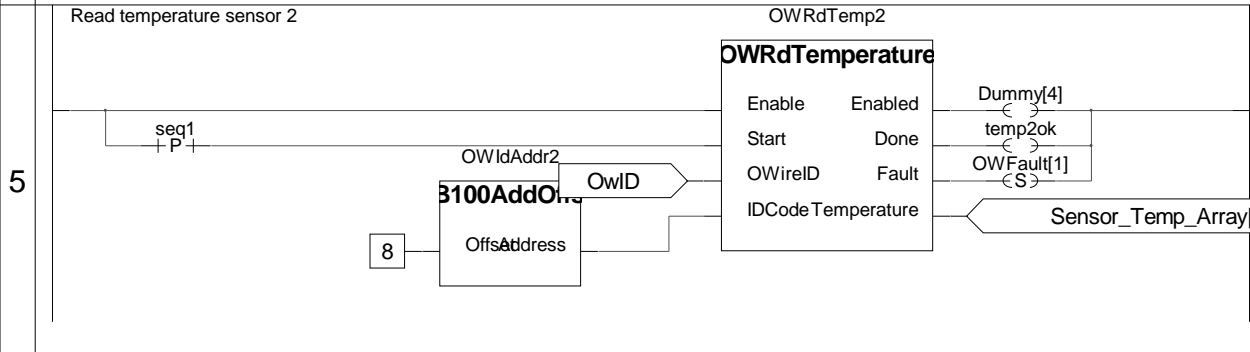
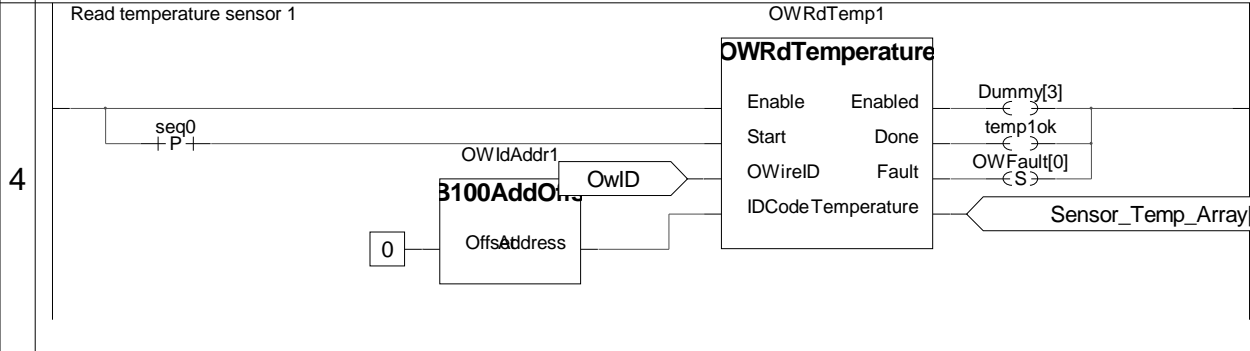
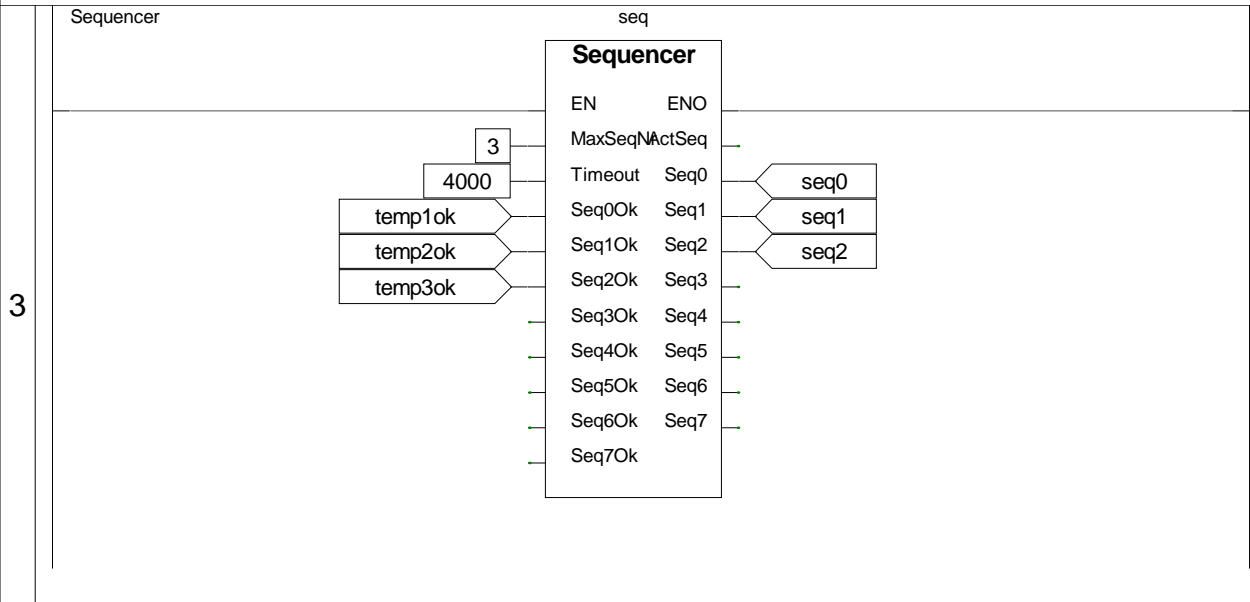
	Project : TempReader	
	VARIABLES :	
	Release :	Ver :1.00
	Author :	Date:02/04/2011
	Note :	Page:1 of 1

```

VAR
Dummy : ARRAY[ 0..7 ] OF BOOL; (* Dummy variables *)
Fp : FILEP; (* File pointer *)
OwID : UDINT; (* One-Wire ID *)
OWidAddr1 : DB100AddOffset;
OWidAddr2 : DB100AddOffset;
OWidAddr3 : DB100AddOffset;
OWire : OWireMng;
OWRdTemp1 : OWRdTemperature;
OWRdTemp2 : OWRdTemperature;
OWRdTemp3 : OWRdTemperature;
seq : Sequencer;
seq0 : BOOL;
seq1 : BOOL;
seq2 : BOOL;
templok : BOOL;
temp2ok : BOOL;
temp3ok : BOOL;
OWFault : ARRAY[ 0..6 ] OF BOOL;
OwMngErr : BOOL;
END_VAR
    
```



	Project : TempReader	
	PROGRAM : LD_ReadTempSeq	
	Release :	Ver :1.00
	Author :	Date:02/04/2011
	Note :	Page:1 of 2



Project : TempReader	
PROGRAM : LD_ReadTempSeq	
Release :	Ver :1.00
Author :	Date:02/04/2011
Note :	Page:2 of 2

PROGRAM ROMIDInit

```
VAR  
i : USINT; (* Auxiliary counter *)  
END_VAR
```

```
1 (* ***** *)  
2 (* ESEGUO COPIA ROM ID DEI SENSORI *)  
3 (* ***** *)  
4  
5 FOR i:=0 TO 7 DO TS0_ROMID[i]:=Sensor0[i]; END_FOR;  
6 FOR i:=0 TO 7 DO TS1_ROMID[i]:=Sensor1[i]; END_FOR;  
7 FOR i:=0 TO 7 DO TS2_ROMID[i]:=Sensor2[i]; END_FOR;  
8  
9 (* [End of file] *)  
10  
11
```

	Project : TempReader	
	PROGRAM : ROMIDInit	
	Release :	Ver :1.00
	Author :	Date:02/04/2011
	Note :	Page:1 of 1

```

VAR_INPUT
MaxSeqNr : USINT; (* Numero max di sequenze gestite *)
Timeout : UDINT; (* Timeout sequenza in mSec *)
Seq0Ok : BOOL; (* Sequenza 0 ok *)
Seq1Ok : BOOL; (* Sequenza 1 ok *)
Seq2Ok : BOOL; (* Sequenza 2 ok *)
Seq3Ok : BOOL; (* Sequenza 3 ok *)
Seq4Ok : BOOL; (* Sequenza 4 ok *)
Seq5Ok : BOOL; (* Sequenza 5 ok *)
Seq6Ok : BOOL; (* Sequenza 6 ok *)
Seq7Ok : BOOL; (* Sequenza 7 ok *)
END_VAR

```

```

VAR_OUTPUT
ActSeq : USINT; (* Sequenza attuale *)
Seq0 : BOOL; (* Sequenza 0 *)
Seq1 : BOOL; (* Sequenza 1 *)
Seq2 : BOOL; (* Sequenza 2 *)
Seq3 : BOOL; (* Sequenza 3 *)
Seq4 : BOOL; (* Sequenza 4 *)
Seq5 : BOOL; (* Sequenza 5 *)
Seq6 : BOOL; (* Sequenza 6 *)
Seq7 : BOOL; (* Sequenza 7 *)
END_VAR

```

```

VAR_EXTERNAL
SysTime : UDINT;
END_VAR

```

```

VAR
time_buff : UDINT;
END_VAR

```

```

1
2 (* Se timeout sequenza ... *)
3 IF((SysTime - time_buff) >= Timeout)THEN
4     ActSeq:=ActSeq+1;
5     time_buff:=SysTime;
6 ELSE
7     (* Controllo se per ogni sequenza c'e' il SeqOk che fa avanzare alla successiva *)
8     CASE ActSeq OF
9         0:
10            IF(Seq0Ok)THEN
11                ActSeq:=ActSeq+1;
12                time_buff:=SysTime;
13            END_IF;
14        1:
15            IF(Seq1Ok)THEN
16                ActSeq:=ActSeq+1;
17                time_buff:=SysTime;
18            END_IF;
19        2:
20            IF(Seq2Ok)THEN
21                ActSeq:=ActSeq+1;
22                time_buff:=SysTime;
23            END_IF;
24        3:

```

Project : TempReader	
FUNCTION BLOCK : Sequencer	
Release :	Ver :1.00
Author :	Date:02/04/2011
Note :	Page:1 of 3

FUNCTION_BLOCK Sequencer

```

25         IF(Seq3Ok) THEN
26             ActSeq:=ActSeq+1;
27             time_buff:=SysTime;
28         END_IF;
29     4:
30         IF(Seq4Ok) THEN
31             ActSeq:=ActSeq+1;
32             time_buff:=SysTime;
33         END_IF;
34     5:
35         IF(Seq5Ok) THEN
36             ActSeq:=ActSeq+1;
37             time_buff:=SysTime;
38         END_IF;
39     6:
40         IF(Seq6Ok) THEN
41             ActSeq:=ActSeq+1;
42             time_buff:=SysTime;
43         END_IF;
44     7:
45         IF(Seq7Ok) THEN
46             ActSeq:=ActSeq+1;
47             time_buff:=SysTime;
48         END_IF;
49     END_CASE ;
50 END_IF;
51
52 (* Clamp su max numero sequenze *)
53 IF(ActSeq >= MaxSeqNr) THEN
54     ActSeq:=0;
55 END_IF;
56
57 (* Gestione bit di uscita *)
58 Seq0:=FALSE;
59 Seq1:=FALSE;
60 Seq2:=FALSE;
61 Seq3:=FALSE;
62 Seq4:=FALSE;
63 Seq5:=FALSE;
64 Seq6:=FALSE;
65 Seq7:=FALSE;
66
67 CASE ActSeq OF
68     0:
69         Seq0:=TRUE;
70     1:
71         Seq1:=TRUE;
72     2:
73         Seq2:=TRUE;
74     3:
75         Seq3:=TRUE;
76     4:
77         Seq4:=TRUE;
78     5:
79         Seq5:=TRUE;
80     6:
81         Seq6:=TRUE;
82     7:
83         Seq7:=TRUE;
84 END_CASE ;

```

Project : TempReader	
FUNCTION BLOCK : Sequencer	
Release :	Ver :1.00
Author :	Date:02/04/2011
Note :	Page:2 of 3

85

86

	Project : TempReader	
	FUNCTION BLOCK : Sequencer	
	Release :	Ver :1.00
	Author :	Date:02/04/2011
	Note :	Page:3 of 3

FUNCTION_BLOCK DB100AddOffset

Returns the DB100 address offset
ENCRYPTED CODE

```
VAR_INPUT  
Offset : UINT; (* Address offset *)  
END_VAR
```

```
VAR_OUTPUT  
Address : UDINT; (* Address value *)  
END_VAR
```

1

	Project : TempReader	
	FUNCTION BLOCK : DB100AddOffset	
	Release :	Ver :1.00
	Author :	Date:02/04/2011
	Note :	Page:1 of 1