

VARIABLES

VAR_GLOBAL

```

WSpeed AT %MD100.256 : ARRAY[ 0..1 ] OF REAL; (* Wind speed (m/s) *)
WorkMode AT %MX100.272 : ARRAY[ 0..1 ] OF USINT; (* Working mode *)
DI00 AT %IX0.0 : BOOL; (* Selezione modo lavoro Ch0 *)
DI01 AT %IX0.1 : BOOL; (* Selezione modo lavoro Ch1 *)
WSColour AT %MX100.264 : ARRAY[ 0..1 ] OF USINT; (* Wind speed colour *)
AlarmOnThr AT %MX100.2048 : ARRAY[ 0..1 ] OF REAL; (* Soglia intervento allarme (m/s) *)
AlarmOnDelay AT %MX100.2056 : ARRAY[ 0..1 ] OF REAL; (* Ritardo intervento allarme (Sec) *)
AlarmBackThr AT %MX100.2064 : ARRAY[ 0..1 ] OF REAL; (* Soglia rientro allarme (m/s) *)
AlarmBackDelay AT %MX100.2072 : ARRAY[ 0..1 ] OF REAL; (* Ritardo rientro allarme (Sec) *)
Do00 AT %QX0.0 : BOOL; (* Relè allarme Ch0 *)
Do01 AT %QX0.1 : BOOL; (* Relè allarme Ch1 *)
END_VAR

```

	Project : WindSpeed	
	VARIABLES :	
	Release : WindSpeed	Ver :1.00
	Author :	Date:07/10/2014
	Note :	Page:1 of 1

```
VAR
AInp : ARRAY[ 0..1 ] OF SysGetAnInp; (* Acquisizione ingresso analogico *)
WSAverage : ARRAY[ 0..1 ] OF Average; (* Wind speed average *)
TimeBf : ARRAY[ 0..1 ] OF UDINT; (* Time buffer (mS) *)
CaseNr : ARRAY[ 0..1 ] OF USINT; (* Case gestione *)
END_VAR
```

```
1 (* ***** *)
2 (* GESTIONE VELOCITA' VENTO *)
3 (* ***** *)
4 (* Eseguo gestione velocità vento. *)
5 (* ----- *)
6
7 (* ----- *)
8 (* INIZIALIZZAZIONI *)
9 (* ----- *)
10 (* Eseguo inizializzazioni. *)
11
12 IF (SysFirstLoop) THEN
13     AInp[0].Address:=0; (* Module address *)
14     AInp[0].Channel:=0; (* Input channel *)
15     AInp[0].Mode:=AD_VOLT_0_10_COMMON; (* Acquisition mode *)
16
17     AInp[1].Address:=0; (* Module address *)
18     AInp[1].Channel:=1; (* Input channel *)
19     AInp[1].Mode:=AD_VOLT_0_10_COMMON; (* Acquisition mode *)
20
21     WSAverage[0].Coefficient:=500.0; (* Average coefficient *)
22     WSAverage[1].Coefficient:=500.0; (* Average coefficient *)
23 END_IF;
24
25 (* ----- *)
26 (* LETTURA VELOCITA' VENTO *)
27 (* ----- *)
28 (* Eseguo acquisizione analogica. *)
29
30 AInp[0](); (* Acquisizione canale 0 *)
31 AInp[1](); (* Acquisizione canale 1 *)
32
33 (* L'anemometro ha un campo di misura da 0,3 ÷ 50 m/s. Quindi con 10 volt *)
34 (* avremo una velocità di 50 m/s. *)
35
36 WSAverage[0].Value:=ValueScale(AInp[0].Value, 0.0, 10.0, 0.0, 50.0); (* Wind speed (m/s) *)
37 WSAverage[1].Value:=ValueScale(AInp[1].Value, 0.0, 10.0, 0.0, 50.0); (* Wind speed (m/s) *)
38
39 (* Eseguo media lettura velocità. *)
40
41 WSAverage[0](); (* Wind speed average channel 0 *)
42 WSAverage[1](); (* Wind speed average channel 1 *)
43 WSpeed[0]:=WSAverage[0].Value; (* Wind speed (m/s) *)
44 WSpeed[1]:=WSAverage[1].Value; (* Wind speed (m/s) *)
45
46 (* ----- *)
47 (* GESTIONE ALLARMI CANALE [0] *)
48 (* ----- *)
49 (* Cases gestione allarmi. *)
50
```

	Project : WindSpeed	
	PROGRAM : WindSpeed	
	Release : WindSpeed	Ver :1.00
	Author :	Date:07/10/2014
	Note :	Page:1 of 2

PROGRAM WindSpeed

```

51 CASE (CaseNr[0]) OF
52
53     (* ----- *)
54     (* Gestione allarme per superamento velocità di soglia. *)
55
56     0:
57     Do00:=FALSE; (* Relè allarme Ch0 *)
58     WSColour[0]:=1; (* Wind speed colour *)
59
60     IF (WSpeed[0] < AlarmOnThr[0]) THEN TimeBf[0]:=SysTime; RETURN; END_IF;
61     IF ((SysTime-TimeBf[0]) < TO_UDINT(AlarmOnDelay[0]*1000.0)) THEN RETURN; END_IF;
62     TimeBf[0]:=SysTime; (* Time buffer (mS) *)
63     CaseNr[0]:=CaseNr[0]+1; (* Case gestione *)
64
65     (* ----- *)
66     (* Gestione allarme per superamento velocità di soglia. *)
67
68     1:
69     Do00:=TRUE; (* Relè allarme Ch0 *)
70     WSColour[0]:=0; (* Wind speed colour *)
71
72     IF (WSpeed[0] > AlarmBackThr[1]) THEN TimeBf[0]:=SysTime; RETURN; END_IF;
73     IF ((SysTime-TimeBf[0]) < TO_UDINT(AlarmBackDelay[0]*1000.0)) THEN RETURN; END_IF;
74     TimeBf[0]:=SysTime; (* Time buffer (mS) *)
75     CaseNr[0]:=0; (* Case gestione *)
76 END_CASE;
77
78 (* [End of file] *)
79

```

	Project : WindSpeed	
	PROGRAM : WindSpeed	
	Release : WindSpeed	Ver :1.00
	Author :	Date:07/10/2014
	Note :	Page:2 of 2

```
VAR
Pulse : ARRAY[ 0..1 ] OF BOOL; (* Ausiliario pulse *)
END_VAR
```

```
1 (* ***** *)
2 (* GESTIONE MODI LAVORO *)
3 (* ***** *)
4 (* Eseguo gestione modi lavoro. *)
5 (* ----- *)
6
7 (* ----- *)
8 (* GESTIONE MODI LAVORO *)
9 (* ----- *)
10 (* Eseguo gestione modi lavoro. Occorre definire nella variabile WorkMode *)
11 (* il valore corrispondente al modo di lavoro selezionato. Il javascript *)
12 (* sulla pagina web visualizzerà il testo relativo. I modi possibili sono *)
13 (* 0:Escluso (Sfondo Rosso). *)
14 (* 1:Automatico (Sfondo Verde). *)
15 (* 2:Manuale On (Sfondo Giallo). *)
16 (* ----- *)
17 (* Con ingresso Di00 eseguo scambio modo lavoro canale 0. *)
18
19 IF (DI00 <> Pulse[0]) THEN
20     Pulse[0]:=DI00; (* Ausiliario pulse *)
21     IF (DI00) THEN
22         WorkMode[0]:=WorkMode[0]+1; (* Working mode *)
23         IF (WorkMode[0] > 2) THEN WorkMode[0]:=0; END_IF;
24     END_IF;
25 END_IF;
26
27 (* Con ingresso Di01 eseguo scambio modo lavoro canale 1. *)
28
29 IF (DI01 <> Pulse[1]) THEN
30     Pulse[1]:=DI01; (* Ausiliario pulse *)
31     IF (DI01) THEN
32         WorkMode[1]:=WorkMode[1]+1; (* Working mode *)
33         IF (WorkMode[1] > 2) THEN WorkMode[1]:=0; END_IF;
34     END_IF;
35 END_IF;
36
37 (* [End of file] *)
38
```

	Project : WindSpeed	
	PROGRAM : WorkingMode	
	Release : WindSpeed	Ver :1.00
	Author :	Date:07/10/2014
	Note :	Page:1 of 1

FUNCTION_BLOCK Average

(ePLCUtyLib_C000) Executes the average over a value
ENCRYPTED CODE

VAR_INPUT
Value : REAL; (* Value to mediate *)
Coefficient : REAL; (* Media coefficient *)
END_VAR

VAR_OUTPUT
Average : REAL; (* Average value *)
END_VAR

1

	Project : WindSpeed	
	FUNCTION BLOCK : Average	
	Release : WindSpeed	Ver :1.00
	Author :	Date:07/10/2014
	Note :	Page:1 of 1

FUNCTION ValueScale

(ePLCUtyLib_C000) Scales a value
ENCRYPTED CODE

```
VAR_INPUT
Input : REAL; (* Input value *)
ILow  : REAL; (* Input low  *)
IHigh : REAL; (* Input high *)
OLow  : REAL; (* Output low *)
OHigh : REAL; (* Output high *)
END_VAR
```

1

	Project : WindSpeed	
	FUNCTION : ValueScale	
	Release : WindSpeed	Ver :1.00
	Author :	Date:07/10/2014
	Note :	Page:1 of 1