

VARIABLES

```
VAR_GLOBAL  
Di00CPU AT %IX255.0 : BOOL; (* Di 00 CPU Module *)  
Di01CPU AT %IX255.1 : BOOL; (* Di 01 CPU Module *)  
Di02CPU AT %IX255.2 : BOOL; (* Di 02 CPU Module *)  
Do00CPU AT %QX255.0 : BOOL; (* Do 00 CPU Module *)  
Do01CPU AT %QX255.1 : BOOL; (* Do 01 CPU Module *)  
END_VAR
```

	Project : DoubleTimeSwitch	
	VARIABLES :	
	Release :	Ver :1.00
	Author :	Date:01/12/2014
	Note :	Page:1 of 1

```

VAR
Day : USINT := 1; (* Set Day *)
Hour : USINT := 10; (* Set Hour *)
HOff : ARRAY[ 0..1 ] OF USINT := [10, 10]; (* Time Off (Hour) *)
HOn : ARRAY[ 0..1 ] OF USINT := [10, 10];
Minute : USINT := 0; (* Set Minute *)
MOff : ARRAY[ 0..1 ] OF USINT := [1, 2]; (* Time Off (Minute) *)
MOn : ARRAY[ 0..1 ] OF USINT := [1, 2];
Month : USINT := 12; (* Set Month *)
Second : USINT := 55; (* Set Second *)
SOff : ARRAY[ 0..1 ] OF USINT := [30, 30]; (* Time Off (Second) *)
SOn : ARRAY[ 0..1 ] OF USINT := [5,10];
Year : UINT := 2014; (* Set Year *)
DateToEpoch : SysDateToETime; (* FB Date to Epoch time *)
EpochToDate : SysETimeToDate; (* FB EpochTime to date *)
END_VAR

```

```

1 (* ***** *)
2 (* MAIN PROGRAM *)
3 (* ***** *)
4
5 (* ----- *)
6 (* GESTIONE IMPOSTAZIONE ORARIO *)
7 (* ----- *)
8 (* Da utilizzarsi per impostare l'orario (E' utile nei test) *)
9
10 IF (Di00CPU) THEN
11     DateToEpoch.Day:=Day;
12     DateToEpoch.Month:=Month;
13     DateToEpoch.Year:=Year;
14     DateToEpoch.Hour:=Hour;
15     DateToEpoch.Minute:=Minute;
16     DateToEpoch.Second:=Second;
17     DateToEpoch();
18     SysDateTime:=DateToEpoch.EpochTime;
19 END_IF;
20
21 (* Usata per visualizzare orario. *)
22
23 EpochToDate(EpochTime:=SysDateTime);
24
25 (* ----- *)
26 (* GESTIONE TEMPORIZZAZIONI *)
27 (* ----- *)
28 (* Controllo se lunedì. *)
29
30 Do00CPU:=FALSE; (* Do 00 CPU Module *)
31 IF (EpochToDate.WeekDay <> 1) THEN Do00CPU:=FALSE; RETURN; END_IF;
32
33 (* Gestione temporizzazione su due orari. *)
34
35 IF (STTimeSwitch(TRUE, SysDateTime, HOn[0], MOn[0], SOn[0], HOff[0], MOff[0], SOff[0])) THEN
36     Do00CPU:=TRUE; (* Do 00 CPU Module *)
37 END_IF;
38
39 IF (STTimeSwitch(TRUE, SysDateTime, HOn[1], MOn[1], SOn[1], HOff[1], MOff[1], SOff[1])) THEN
40     Do00CPU:=TRUE; (* Do 00 CPU Module *)

```

Project : DoubleTimeSwitch	
PROGRAM : ST	
Release : DoubleTime	Ver : 1.00
Author :	Date: 01/12/2014
Note :	Page: 1 of 2

PROGRAM ST

```
41     END_IF;  
42  
43 (* [End of file] *)  
44
```

	Project : DoubleTimeSwitch	
	PROGRAM : ST	
	Release : DoubleTime	Ver :1.00
	Author :	Date:01/12/2014
	Note :	Page:2 of 2

FUNCTION STTimeSwitch

```

VAR_INPUT
Enable : BOOL; (* FB Enable *)
DateTime : UDINT; (* Date/Time (Epoch) *)
HOn : USINT; (* Time On (Hour) *)
MOn : USINT; (* Time On (Minute) *)
SOn : USINT; (* Time On (Second) *)
HOff : USINT; (* Time Off (Hour) *)
MOff : USINT; (* Time Off (Minute) *)
SOff : USINT; (* Time Off (Second) *)
END_VAR

VAR
DSEcs : UDINT; (* Day seconds *)
END_VAR
    
```

```

1 (* ***** *)
2 (* FUNCTION "STTimeSwitch" *)
3 (* ***** *)
4 (* Questa FB controlla gli orari impostati ed attiva l'uscita se ora è nel *)
5 (* range definito. *)
6 (* ----- *)
7
8 (* ----- *)
9 (* GESTIONE TEMPORIZZAZIONE *)
10 (* ----- *)
11 (* Controllo se funzione abilitata. *)
12
13 IF NOT(Enable) THEN STTimeSwitch:=FALSE; RETURN; END_IF;
14 DSEcs:=MOD(DateTime, 86400); (* Day seconds*)
15 STTimeSwitch:=(DSEcs >= ((TO_UDINT(HOn)*3600)+(TO_UDINT(MOn)*60)+SOn)) AND (DSEcs < ((TO_UDINT(HOff)
) *3600)+(TO_UDINT(MOff)*60)+SOff));
16
17 (* [End of file] *)
18
19
    
```

	Project : DoubleTimeSwitch	
	FUNCTION : STTimeSwitch	
	Release : DoubleTime	Ver :1.00
	Author :	Date:01/12/2014
	Note :	Page:1 of 1