```
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 *)
Sp AT %MX100.2048 : ARRAY[ 0..6 ] OF DAYDEFINITION; (* Set point *)
END_VAR
```

```
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 *)
Wd : USINT; (* Week day *)
END_VAR
```

```
(* ********************************************************************** *)
(* MAIN PROGRAM                                                           *)
(* ********************************************************************** *)

    (* ---------------------------------------------------------------------- *)
    (* GESTIONE IMPOSTAZIONE ORARIO                                           *)
    (* ---------------------------------------------------------------------- *)
    (* Da utilizzarsi per impostare l'orario (E' utile nei test) *)

    IF (Di00CPU) THEN
        DateToEpoch.Day:=Day;
        DateToEpoch.Month:=Month;
        DateToEpoch.Year:=Year;
        DateToEpoch.Hour:=Hour;
        DateToEpoch.Minute:=Minute;
        DateToEpoch.Second:=Second;
        DateToEpoch();
        SysDateTime:=DateToEpoch.EpochTime;
    END_IF;

    (* Usata per visualizzare orario. *)

    EpochToDate(EpochTime:=SysDateTime);

    (* ---------------------------------------------------------------------- *)
    (* GESTIONE TEMPORIZZAZIONI                                               *)
    (* ---------------------------------------------------------------------- *)
    (* Controllo se lunedì. *)

    Do00CPU:=FALSE; (* Do 00 CPU Module *)
    Do01CPU:=FALSE; (* Do 01 CPU Module *)
    Wd:=EpochToDate.WeekDay; (* Week day *)

    (* Gestione temporizzazione su due orari. *)

    IF (STTimeSwitch(TRUE, SysDateTime, Sp[Wd].Zone[0].Prg[0].HOn, Sp[Wd].Zone[0].Prg[0].MOn, Sp[Wd].Zo
ne[0].Prg[0].SOn, Sp[Wd].Zone[0].Prg[0].HOff, Sp[Wd].Zone[0].Prg[0].MOff, Sp[Wd].Zone[0].Prg[0].SOff))
THEN
        Do00CPU:=TRUE; (* Do 00 CPU Module *)
```

```
38      END_IF;
39
40      IF (STTimeSwitch(TRUE, SysDateTime, Sp[Wd].Zone[0].Prg[1].HOn, Sp[Wd].Zone[0].Prg[1].MOn, Sp[Wd].Zo
    ne[0].Prg[1].SOn, Sp[Wd].Zone[0].Prg[1].HOff, Sp[Wd].Zone[0].Prg[1].MOff, Sp[Wd].Zone[0].Prg[1].SOff))
    THEN
41          Do00CPU:=TRUE; (* Do 00 CPU Module *)
42      END_IF;
43
44      IF (STTimeSwitch(TRUE, SysDateTime, Sp[Wd].Zone[1].Prg[0].HOn, Sp[Wd].Zone[1].Prg[0].MOn, Sp[Wd].Zo
    ne[1].Prg[0].SOn, Sp[Wd].Zone[1].Prg[0].HOff, Sp[Wd].Zone[1].Prg[0].MOff, Sp[Wd].Zone[1].Prg[0].SOff))
    THEN
45          Do01CPU:=TRUE; (* Do 01 CPU Module *)
46      END_IF;
47
48      IF (STTimeSwitch(TRUE, SysDateTime, Sp[Wd].Zone[1].Prg[1].HOn, Sp[Wd].Zone[1].Prg[1].MOn, Sp[Wd].Zo
    ne[1].Prg[1].SOn, Sp[Wd].Zone[1].Prg[1].HOff, Sp[Wd].Zone[1].Prg[1].MOff, Sp[Wd].Zone[1].Prg[1].SOff))
    THEN
49          Do01CPU:=TRUE; (* Do 01 CPU Module *)
50      END_IF;
51
52  (* [End of file] *)
53
54
```

FUNCTION STTimeSwitch

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

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

```
(* ********************************************************************** *)
(* FUNCTION "STTimeSwitch"                                                *)
(* ********************************************************************** *)
(* Questa FB controlla gli orari impostati ed attiva l'uscita se ora è nel    *)
(* range definito.                                                        *)
(* ---------------------------------------------------------------------- *)

    (* ---------------------------------------------------------------------- *)
    (* GESTIONE TEMPORIZZAZIONE                                               *)
    (* ---------------------------------------------------------------------- *)
    (* Controllo se funzione abilitata. *)

    IF NOT(Enable) THEN STTimeSwitch:=FALSE; RETURN; END_IF;
    DSecs:=MOD(DateTime, 86400); (* Day seconds*)
    STTimeSwitch:=(DSecs >= ((TO_UDINT(HOn)*3600)+(TO_UDINT(MOn)*60)+SOn)) AND (DSecs < ((TO_UDINT(HOff)*3600)+(TO_UDINT(MOff)*60)+SOff));

(* [End of file] *)
```

	Project : TimeSwWithStruct	
	FUNCTION : STTimeSwitch	
	Release : TimeSwWith	Ver :1.00
	Author :	Date:02/12/2014
	Note :	Page:1 of 1