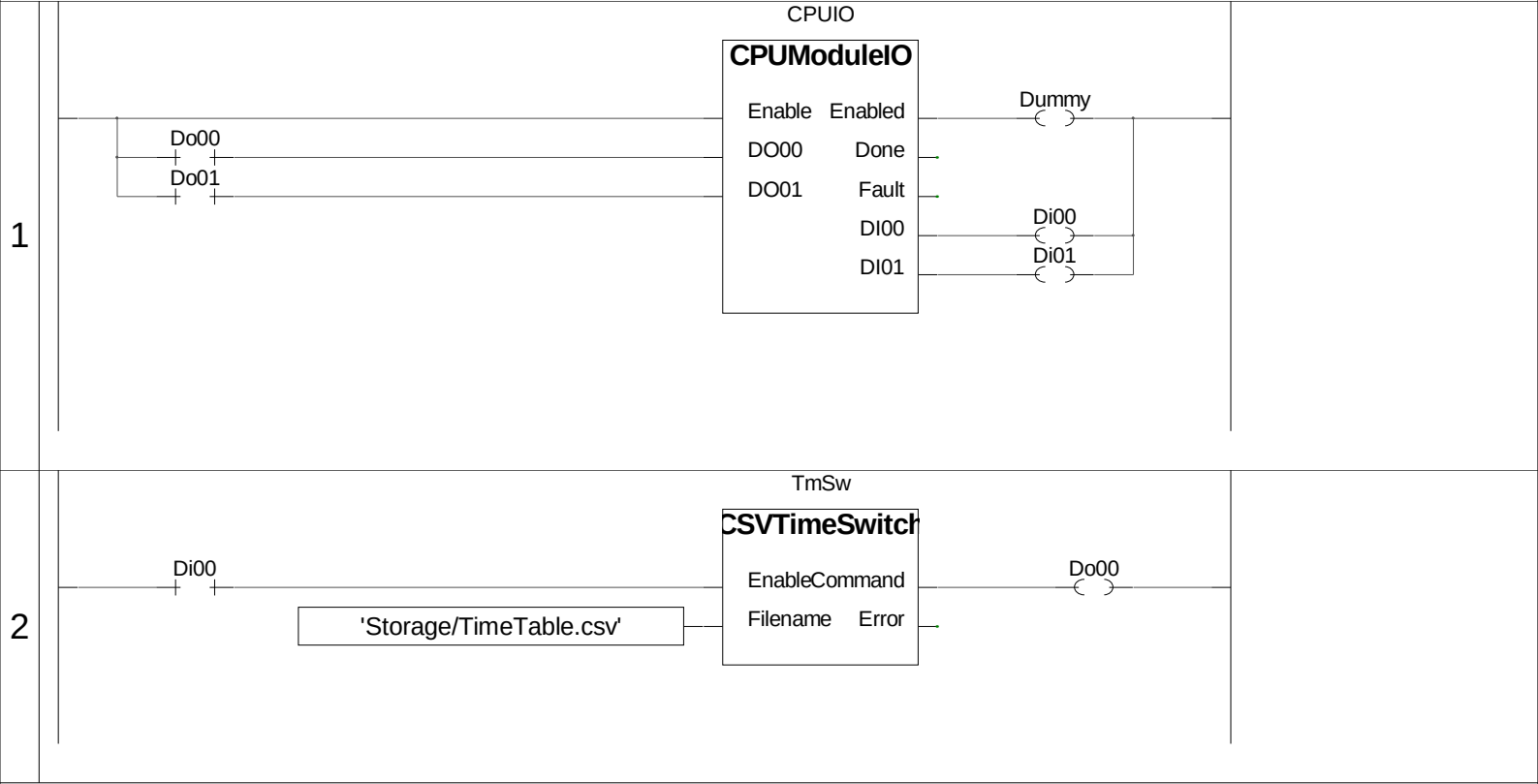


```
VAR
TmSw : CSVTimeSwitch; (* FB CSVTimeSwitch data *)
Dummy : BOOL; (* Dummy variable *)
CPUIO : CPUModuleIO; (* FB CPUIO data *)
Di00 : BOOL; (* Digital Input 0 on CPU module *)
Di01 : BOOL; (* Digital Input 1 on CPU module *)
Do00 : BOOL; (* Digital Output 0 on CPU module *)
Do01 : BOOL; (* Digital Output 1 on CPU module *)
END_VAR
```



2

TmSw

CSVTimeSwitch

EnableCommand

Filename

Error

Di00

'Storage/TimeTable.csv'

Do00

|  |                             |                 |
|--|-----------------------------|-----------------|
|  | Project : CSVFileTimeSwitch |                 |
|  | PROGRAM : LD                |                 |
|  | Release :                   | Ver :1.00       |
|  | Author :                    | Date:29/08/2011 |
|  | Note :                      | Page:1 of 1     |

## VAR\_INPUT

```
Enable : BOOL; (* FB Enable *)
Filename : STRING[ 32 ]; (* Time switch file name *)
END_VAR
```

## VAR\_OUTPUT

```
Command : BOOL; (* Time command *)
Error : USINT; (* Error code *)
END_VAR
```

## VAR\_EXTERNAL

```
SysDateTime : UDINT; (* System Date/Time (Epoch) *)
SysClock1000 : BOOL; (* Pulse every 1 Second *)
END_VAR
```

## VAR

```
Pulse : BOOL; (* Time base pulse *)
CaseBack : USINT; (* Case back *)
CaseNr : USINT; (* Case number *)
SSeq : USINT; (* String sequencer *)
Ch : INT; (* Character read *)
i : INT; (* Auxiliary counter *)
FPos : DINT; (* File position *)
PrOffTime : STRING[ 16 ]; (* Preset off time definition *)
PrOnTime : STRING[ 16 ]; (* Preset on time definition *)
PrWeekDay : STRING[ 16 ]; (* Preset day of week definition *)
Fp : FILEP; (* File pointer *)
Ptr : @USINT; (* Internal pointer *)
SArray : ARRAY[ 0..2 ] OF USINT; (* Sting scanf value array *)
WeekDay : USINT; (* Day of week *)
DateTime : SysTimeToDate; (* Date/Time value *)
TimeAbs : UDINT; (* Time absolute value (Sec) *)
END_VAR
```

```
1 (* ***** *)
2 (* FB GESTIONE PROGRAMMATTORE ORARIO *)
3 (* ***** *)
4 (* Questo blocco funzione esegue lettura degli orari accensione/spegnimento *)
5 (* da un file CSV e gestisce il comando in uscita. Il file CSV deve iniziare *)
6 (* con un record definizione campi e poi seguire con i record di definizione *)
7 (* degli orari. E' possibile definire più di un orario per giorno, gli orari *)
8 (* devono essere definiti in ordine cronologico a partire dal minore. *)
9 (* Se si vuole gestire definizione oraria che inizia in un giorno e termina *)
10 (* nel giorno successivo occorre definire orario di spegnimento "24:00:00" ed *)
11 (* orario di accensione al giorno successivo "00:00:00". *)
12 (* *)
13 (* "Giorno","Accensione","Spegnimento" *)
14 (* "Monday","10:20:00","10:20:10" *)
15 (* "Monday","15:30:00","16:35:00" *)
16 (* "Tuesday","08:05:10","09:15:30" *)
17 (* ----- *)
18
19 (* ----- *)
20 (* GESTIONE ABILITAZIONE *)
21 (* ----- *)
22 (* Eseguo gestione abilitazione. *)
23
```

Project : CSVFileTimeSwitch

FUNCTION BLOCK : CSVTimeSwitch

Release : CSVFileTim

Ver :1.00

Author :

Date:29/08/2011

Note :

Page:1 of 5

```

24 IF NOT(Enable) THEN Command:=FALSE; RETURN; END_IF;
25
26 (* Eseguo la gestione ad ogni secondo. *)
27
28 IF (SysClock1000 = Pulse) THEN RETURN; END_IF;
29 Pulse:=SysClock1000; (* Time base pulse *)
30
31 (* ----- *)
32 (* LETTURA FILE DEFINIZIONE ORARI *)
33 (* ----- *)
34 (* Apro file in "read", indicatore di posizione stream punta inizio. *)
35
36 Fp:=Sysfopen(Filename, 'r'); (* File pointer *)
37 IF (Fp = NULL) THEN Error:=10; RETURN; END_IF;
38
39 (* Eseguo calcolo data/ora da Epoch time. *)
40
41 DateTime.EpochTime:=SysDateTime;
42 DateTime();
43 TimeAbs:=(DateTime.Hour*3600)+(DateTime.Minute*60)+DateTime.Second; (* Time absolute value (Sec) *)
44
45 (* ----- *)
46 (* GESTIONE LOOP *)
47 (* ----- *)
48 (* Eseguo loop chiuso su while per permettere il passaggio di case senza *)
49 (* uscire dal programma. *)
50
51 CaseNr:=0; (* Case number *)
52 WHILE (TRUE) DO
53
54     (* ----- *)
55     (* GESTIONE NUMERO DI CASE *)
56     (* ----- *)
57     (* Eseguo gestione chiamate subroutines di case. *)
58
59     CASE (CaseNr) OF
60
61         (* ##### *)
62         (* CASES ELIMINAZIONE INTESTAZIONE FILE CSV *)
63         (* ##### *)
64
65         (* ----- *)
66         (* La prima riga file contiene definizione campi, la scarto. *)
67         (* "Giorno","Accensione","Spegnimento" *)
68
69         0:
70         CaseBack:=10; (* Case back *)
71         CaseNr:=100; (* Case number *)
72
73         (* ##### *)
74         (* CASES GESTIONE COMANDO ORARIO *)
75         (* ##### *)
76
77         (* ----- *)
78         (* Eseguo lettura record definizione tempi. *)
79         (* "Monday","10:20:00","10:21:01" *)
80
81         10:
82         CaseBack:=11; (* Case back *)
83         CaseNr:=100; (* Case number *)

```

Project : CSVFileTimeSwitch

FUNCTION BLOCK : CSVTimeSwitch

Release : CSVFileTim

Ver :1.00

Author :

Date:29/08/2011

Note :

Page:2 of 5

# FUNCTION\_BLOCK CSVTimeSwitch

```

84
85      (* ----- *)
86      (* Calcolo valore giorno della settimana. *)
87
88      11:
89      IF (eFIND(ADR(PrWeekDay), ADR('Sunday')) <> 0) THEN WeekDay:=0;
90      ELSIF (eFIND(ADR(PrWeekDay), ADR('Monday')) <> 0) THEN WeekDay:=1;
91      ELSIF (eFIND(ADR(PrWeekDay), ADR('Tuesday')) <> 0) THEN WeekDay:=2;
92      ELSIF (eFIND(ADR(PrWeekDay), ADR('Wednesday')) <> 0) THEN WeekDay:=3;
93      ELSIF (eFIND(ADR(PrWeekDay), ADR('Thursday')) <> 0) THEN WeekDay:=4;
94      ELSIF (eFIND(ADR(PrWeekDay), ADR('Friday')) <> 0) THEN WeekDay:=5;
95      ELSIF (eFIND(ADR(PrWeekDay), ADR('Saturday')) <> 0) THEN WeekDay:=6;
96      ELSE Error:=20; EXIT;
97      END_IF;
98
99      (* Se giorno settimana non coincide con giorno attuale, passo a *)
100     (* controllare record orario successivo da file CSV. *)
101
102     IF (WeekDay <> DateTime.WeekDay) THEN CaseNr:=10;(* Case number *)
103     ELSE CaseNr:=CaseNr+1; (* Case number *)
104     END_IF;
105
106     (* ----- *)
107     (* Lettura campi tempo di accensione e calcolo valore assoluto. *)
108
109     12:
110     IF NOT(SysVarsscanf(ADR(PrOnTime)+0, '%2d', USINT_TYPE, ADR(SArray[0]))) THEN Error:=30; EX
IT; END_IF;
111     IF NOT(SysVarsscanf(ADR(PrOnTime)+3, '%2d', USINT_TYPE, ADR(SArray[1]))) THEN Error:=31; EX
IT; END_IF;
112     IF NOT(SysVarsscanf(ADR(PrOnTime)+6, '%2d', USINT_TYPE, ADR(SArray[2]))) THEN Error:=32; EX
IT; END_IF;
113
114     (* Controllo se orario attuale superiore ad orario accensione. *)
115
116     IF (TimeAbs >= (SArray[0]*3600)+(SArray[1]*60)+SArray[2]) THEN Command:=TRUE; END_IF;
117
118     (* Lettura campi tempo di spegnimento e calcolo valore assoluto. *)
119
120     IF NOT(SysVarsscanf(ADR(PrOffTime)+0, '%2d', USINT_TYPE, ADR(SArray[0]))) THEN Error:=40; E
XIT; END_IF;
121     IF NOT(SysVarsscanf(ADR(PrOffTime)+3, '%2d', USINT_TYPE, ADR(SArray[1]))) THEN Error:=41; E
XIT; END_IF;
122     IF NOT(SysVarsscanf(ADR(PrOffTime)+6, '%2d', USINT_TYPE, ADR(SArray[2]))) THEN Error:=42; E
XIT; END_IF;
123
124     (* Controllo se orario attuale superiore ad orario spegnimento. *)
125
126     IF (TimeAbs >= (SArray[0]*3600)+(SArray[1]*60)+SArray[2]) THEN Command:=FALSE; END_IF;
127
128     (* Passo a controllare record orario successivo da file CSV. *)
129
130     CaseNr:=10;(* Case number *)
131
132     (* ##### *)
133     (* LETTURA RECORD DA FILE CSV *)
134     (* ##### *)
135
136     (* ----- *)
137     (* Lettura valore di giorno della settimana dal file CSV. *)

```

|  |                                |                 |
|--|--------------------------------|-----------------|
|  | Project : CSVFileTimeSwitch    |                 |
|  | FUNCTION BLOCK : CSVTimeSwitch |                 |
|  | Release : CSVFileTim           | Ver :1.00       |
|  | Author :                       | Date:29/08/2011 |
|  | Note :                         | Page:3 of 5     |

```

138
139      100:
140      SSeq:=0; (* String sequencer *)
141      Ptr:=ADR(PrWeekDay); (* Internal pointer *)
142      i:=MemSet(Ptr, 0, 16); (* Azzerò buffer stringa *)
143      CaseNr:=CaseNr+1; (* Case number *)
144
145      (* ----- *)
146      (* Lettura valore tempo di accensione dal file CSV. *)
147
148      102:
149      SSeq:=0; (* String sequencer *)
150      Ptr:=ADR(PrOnTime); (* Internal pointer *)
151      i:=MemSet(Ptr, 0, 16); (* Azzerò buffer stringa *)
152      CaseNr:=CaseNr+1; (* Case number *)
153
154      (* ----- *)
155      (* Lettura valore tempo di spegnimento dal file CSV. *)
156
157      104:
158      SSeq:=0; (* String sequencer *)
159      Ptr:=ADR(PrOffTime); (* Internal pointer *)
160      i:=MemSet(Ptr, 0, 16); (* Azzerò buffer stringa *)
161      CaseNr:=CaseNr+1; (* Case number *)
162
163      (* ----- *)
164      (* Arrivo con il puntatore al file che punta <CR><LF>. *)
165
166      106:
167      IF (Sysfgetc(Fp) <> 13) THEN Error:=200; EXIT; END_IF;
168      IF (Sysfgetc(Fp) <> 10) THEN Error:=201; EXIT; END_IF;
169      CaseNr:=CaseBack; (* Case number *)
170
171      (* ----- *)
172      (* Viene eseguita lettura di un campo stringa dal record dati su *)
173      (* file. Ogni campo stringa è racchiuso tra '"' Il record dati *)
174      (* termina con il <LF>. *)
175
176      101, 103, 105:
177      Ch:=Sysfgetc(Fp); (* Character read *)
178      IF (Ch = EOF) THEN EXIT; END_IF;
179
180      CASE (SSeq) OF
181
182          (* ----- *)
183          (* Attesa '"' di inizio stringa. *)
184
185          0:
186          IF (Ch = 34) THEN SSeq:=SSeq+1; END_IF;
187
188          (* ----- *)
189          (* Lettura valore stringa fino a '"' di fine. *)
190
191          1:
192          IF (Ch = 34) THEN CaseNr:=CaseNr+1; (* Case number *)
193          ELSE @Ptr:=TO_USINT(Ch); Ptr:=Ptr+1; END_IF;
194      ELSE
195          Error:=210; (* Error code *)
196          EXIT;
197      END_CASE;

```

Project : CSVFileTimeSwitch

FUNCTION BLOCK : CSVTimeSwitch

Release : CSVFileTim

Ver :1.00

Author :

Date:29/08/2011

Note :

Page:4 of 5

# FUNCTION\_BLOCK CSVTimeSwitch

```

198
199      (* ----- *)
200      (* GESTIONE CASE ERRATO *)
201      (* ----- *)
202
203      ELSE
204          Error:=250; (* Error code *)
205          EXIT;
206      END_CASE;
207  END_WHILE;
208
209      (* Eseguo chiusura file. *)
210
211      IF (Sysfclose(Fp) <> 0) THEN Error:=251; END_IF;
212
213  (* [End of file] *)
214
215

```

|  |                                |                 |
|--|--------------------------------|-----------------|
|  | Project : CSVFileTimeSwitch    |                 |
|  | FUNCTION BLOCK : CSVTimeSwitch |                 |
|  | Release : CSVFileTim           | Ver :1.00       |
|  | Author :                       | Date:29/08/2011 |
|  | Note :                         | Page:5 of 5     |

Manages the logic I/O on the CPU module  
ENCRYPTED CODE

```
VAR_INPUT
Enable : BOOL; (* Function enable *)
DO00 : BOOL; (* Digital output 0 *)
DO01 : BOOL; (* Digital output 1 *)
END_VAR

VAR_OUTPUT
Enabled : BOOL; (* Function enabled *)
Done : BOOL; (* Function done *)
Fault : BOOL; (* Function fault *)
DI00 : BOOL; (* Digital input 0 *)
DI01 : BOOL; (* Digital input 0 *)
END_VAR
```

1

Project : CSVFileTimeSwitch

FUNCTION BLOCK : CPUModuleIO

Release : CSVFileTim

Ver :1.00

Author :

Date:29/08/2011

Note :

Page:1 of 1

## FUNCTION MemSet

Fills memory with value  
ENCRYPTED CODE

```
VAR_INPUT
Buffer : @USINT; (* Buffer address *)
Value : USINT; (* Value to set *)
Size : UINT; (* Buffer size *)
END_VAR
```

1

Project : CSVFileTimeSwitch

FUNCTION : MemSet

Release : CSVFileTim

Ver :1.00

Author :

Date:29/08/2011

Note :

Page:1 of 1



String find  
ENCRYPTED CODE

```
VAR_INPUT
In1 : @USINT; (* String where looking *)
In2 : @USINT; (* String to find *)
END_VAR
```

1

|  |                             |                 |
|--|-----------------------------|-----------------|
|  | Project : CSVFileTimeSwitch |                 |
|  | FUNCTION : eFIND            |                 |
|  | Release : CSVFileTim        | Ver :1.00       |
|  | Author :                    | Date:29/08/2011 |
|  | Note :                      | Page:1 of 1     |

String length calculation  
ENCRYPTED CODE

```
VAR_INPUT
Str : @USINT; (* String pointer *)
END_VAR
```

1

|  |                             |                 |
|--|-----------------------------|-----------------|
|  | Project : CSVFileTimeSwitch |                 |
|  | FUNCTION : eLEN             |                 |
|  | Release : CSVFileTim        | Ver :1.00       |
|  | Author :                    | Date:29/08/2011 |
|  | Note :                      | Page:1 of 1     |