```
VAR
Pulse : BOOL; (* Time base pulse *)
NrOfLog : USINT; (* Number of log stored *)
i : INT; (* Auxiliary counter *)
LogRecord : STRING[ 64 ]; (* Log record *)
Fp : FILEP; (* File pointer *)
DateTime : SysETimeToDate; (* Date/Time conversion *)
TimeBf : UDINT; (* Time buffer (uS) *)
WDays : ARRAY[ 0..6 ] OF WEEKDAY; (* Giorni settimanali *)
Filename : STRING[ 32 ]; (* Nome file *)
END_VAR
```

```
(* ************************************************************************** *)
(* WRITE A VOLTAGE LOG                                                        *)
(* ************************************************************************** *)
(* Ogni 10 secondi viene scritto un record nel file di log. Il record scritto *)
(* è del tipo "2014-10-08 09:25:58;02.8;02.8;02.8;02.8;".                     *)
(* -------------------------------------------------------------------------- *)

    (* ---------------------------------------------------------------------- *)
    (* INIZIALIZZAZIONE                                                       *)
    (* ---------------------------------------------------------------------- *)
    (* Eseguo inizializzazione. *)

    IF (SysFirstLoop) THEN

        (* Eseguo definizione dei giorni settimanali. *)

        WDays[0].Name:='Domenica';
        WDays[1].Name:='Lunedi';
        WDays[2].Name:='Martedi';
        WDays[3].Name:='Mercoledi';
        WDays[4].Name:='Giovedi';
        WDays[5].Name:='Venerdi';
        WDays[6].Name:='Sabato';

        (* Salvo riferimento tempo per temporizzazione. *)

        TimeBf:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
    END_IF;

    (* ---------------------------------------------------------------------- *)
    (* BASE TEMPI REGISTRAZIONE                                               *)
    (* ---------------------------------------------------------------------- *)
    (* Gestisco base tempi registrazione. *)

    IF ((SysGetSysTime(TRUE)-TimeBf) < 10000000) THEN RETURN; END_IF;
    TimeBf:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
    DateTime(EpochTime:=SysDateTime); (* Date time conversion *)

    (* Creo nome file in base al giorno settimanale. *)

    Filename:=CONCAT('SDCard/', WDays[DateTime.WeekDay].Name); (* Nome file *)
    Filename:=CONCAT(Filename, '.csv'); (* Nome file *)

    (* ---------------------------------------------------------------------- *)
    (* DEFINIZIONE FILE DI LOG                                                *)
```

```
    (* ------------------------------------------------------------------- *)
    (* Controllo se mese attuale diverso da mese in log. *)

    IF (DateTime.WeekDay <> LogWkDay) THEN
        LogWkDay:=DateTime.WeekDay; (* Giorno settimana in log *)

        (* Eseguo cancellazione del vecchio file di log. Si tratta del file *)
        (* relativo alla settimana precedente. *)

        IF (Sysfilelength(Filename) <> EOF) THEN
            i:=Sysremove(Filename); (* Eseguo cancellazione file *)
        END_IF;
    END_IF;

    (* ------------------------------------------------------------------- *)
    (* WRITE THE LOG                                                       *)
    (* ------------------------------------------------------------------- *)
    (* Open the file in "append" mode. *)

    Fp:=Sysfopen(Filename, 'a'); (* File pointer *)
    IF (Fp = NULL) THEN RETURN; END_IF;

    (* Initialize the log record. *)

    i:=MemSet(ADR(LogRecord), 0, 32);

    (* Please note that any value is written starting at the position of the *)
    (* "0" string terminator of the previous value overwriting it. *)

    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 6, '%04d-', UINT_TYPE, ADR(DateTime.Year));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 4, '%02d-', USINT_TYPE, ADR(DateTime.Month));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 4, '%02d ', USINT_TYPE, ADR(DateTime.Day));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 4, '%02d:', USINT_TYPE, ADR(DateTime.Hour));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 4, '%02d:', USINT_TYPE, ADR(DateTime.Minute));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 4, '%02d;', USINT_TYPE, ADR(DateTime.Second));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 6, '%04.1f;', REAL_TYPE, ADR(Voltage));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 6, '%04.1f;', REAL_TYPE, ADR(Voltage));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 6, '%04.1f;', REAL_TYPE, ADR(Voltage));
    i:=SysVarsnprintf(ADR(LogRecord)+LEN(LogRecord), 8, '%04.1f;$r$n', REAL_TYPE, ADR(Voltage));

    (* Eseguo scrittura record nel file. *)

    i:=Sysfwrite(ADR(LogRecord), TO_INT(LEN(LogRecord)), 1, Fp); (* Write to file *)
    i:=Sysfclose(Fp); (* Close file *)

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