

```

VAR_INPUT
Filename : STRING[ 32 ]; (* Percorso e nome del file *)
Value : REAL; (* Valore da memorizzare *)
END_VAR

VAR_OUTPUT
Error : UDINT; (* Error number *)
END_VAR

VAR
i : UDINT; (* Auxiliary variable *)
Fp : FILEP; (* File pointer *)
FExists : BOOL; (* File esiste *)
END_VAR

```

```

1 // *****
2 // FUNCTION BLOCK "FileStore"
3 // *****
4 // Memorizzazione dato su file.
5 //
6 // Parametri in ingresso
7 // Value: Valore da memorizzare
8 // Filename: Percorso e nome del file
9 // .
10 // -----
11
12 // -----
13 // CONTROLLO SE FILE ESISTE
14 // -----
15 // Se file non esiste è il primo dato scritto quindi scrivo solo valore,
16 // se il file esiste occorre inserire il ";" di separazione valori.
17
18 FExists:=TO_BOOL(Sysfilelength(Filename) <> EOF); //File esiste
19
20 // -----
21 // ESEGUO APERTURA FILE
22 // -----
23 // Apertura file in append, se il file non esiste viene creato, se esiste
24 // ci si posiziona alla fine.
25
26 Fp:=Sysfopen(Filename, 'a'); (* File pointer *)
27 IF (Fp = NULL) THEN Error:=50; RETURN; END_IF;
28
29 // -----
30 // SCRITTURA VALORE
31 // -----
32 // Il valore viene scritto con 2 cifre intere ed 1 decimale.
33
34
35 IF NOT(FExists) THEN
36   i:=SysVarfprintf(Fp, '%.1f', REAL_TYPE, ADR(Value));
37 ELSE
38   i:=SysVarfprintf(Fp, ';' &%.1f', REAL_TYPE, ADR(Value));
39 END_IF;
40
41 // Eseguo chiusura file.
42
43 i:=Sysfclose(Fp); //Eseguo chiusura file
44
45 (* [End of file] *)
46

```

Project : SimplyLogger

FUNCTION_BLOCK : FileStore

Release : SimplyLogg

Ver :1.00

Author :

Date:12/02/2019

Note :

Page:1 of 1

```

VAR
TimeBf : UDINT; (* Time buffer (uS) *)
AInput : SysGetAnInp; (* Analog input acquisition *)
FStore : FileStore; (* Memorizzazione su file *)
DateTime : SysTimeToDate; (* Data ed ora attuali *)
Hour : USINT; (* Ora riferimento *)
i : UDINT; (* Auxiliary variable *)
END_VAR

```

```

1 // *****
2 // PROGRAM "DataAcquisition"
3 // *****
4 // Esego acquisizione dati.
5 // -----
6
7 // -----
8 // INIZIALIZZAZIONE
9 // -----
10 // Esego inizializzazione.
11
12 IF (SysFirstLoop) THEN
13   AInput.Address:=255; //Module address
14   AInput.Address:=0; //Module address
15   AInput.Channel:=0; //Input channel
16   AInput.Mode:=AD_VOLT_0_10_COMMON; //Acquisition mode
17
18   TimeBf:=SysGetSysTime(TRUE); //Time buffer (uS)
19 END_IF;
20
21 // -----
22 // ESEGUO GESTIONE DATA/ORA
23 // -----
24 // Gestione data/ora attuali.
25
26 DateTime(EpochTime:=SysDateTime); //Data ed ora attuali
27
28 // -----
29 // TEMPO DI CAMPIONAMENTO VALORI
30 // -----
31 // Esego controllo su tempo campionamento.
32
33 IF ((SysGetSysTime(TRUE)-TimeBf) < 300000) THEN RETURN; END_IF;
34 TimeBf:=SysGetSysTime(TRUE); //Time buffer (uS)
35
36 // Esego acquisizione analogica.
37
38 AInput(); //Acquisizione analogica
39
40 // -----
41 // MEMORIZZO DATI SU FILE
42 // -----
43 // Gestione nome file, ad ogni cambio di ora viene creato un file con nome
44 // GGMHHH.cvv (Esempio D:/120200.csv).
45
46 IF ((LEN(FStore.Filename) = 0) OR (DateTime.Hour <> Hour)) THEN
47   Hour:=DateTime.Hour; //Ora riferimento
48
49   // Creo nome file
50
51   i:=SysVarsnprintf(ADR(FStore.Filename), sizeof(FStore.Filename), 'D:%02d', USINT_TYPE, ADR(DateTime.Day));
52   i:=SysLWVarsnprintf(ADR(FStore.Filename), sizeof(FStore.Filename), '%02d', USINT_TYPE, ADR(DateTime.Month));
53   i:=SysLWVarsnprintf(ADR(FStore.Filename), sizeof(FStore.Filename), '%02d.csv', USINT_TYPE, ADR(DateTime.Hour));
54 END_IF;
55
56 // Esego memorizzazione su file.
57
58 FStore.Value:=AInput.Value; //Valore da memorizzare
59 FStore(); //Memorizzazione su file
60
61 (* [End of file] *)
62

```

Project : SimplyLogger

PROGRAM : DataAcquisition

Release : SimplyLogg

Ver :1.00

Author :

Date:12/02/2019

Note :

Page:1 of 1