LoggerSI library
User Guide

Version 1.2.2

Contents

Library Contents 4

Library description 4

Black Box 5

FB_BlackBox 5

Cloud 8

CharlieJsonFormatGenerator 8

GetTimeUTC 9

MqttIoTHub4Insight 9

PubMQTTmsgFromFile 10

DataLogger 12

LogFile12iData 12

LogFileGenericData 16

Files utilities 18

AddStringToFile 18

CreationFilesList 19

ExportAllToUSB 20

FileNameCreation 20

Structures 22

strAsset 22

strDataAsset 22

strLogFileEvent 23

strLogFileGenSignedData 23

strString 23

LOGGERSI_CONSTS 24

Safety Information 25

Library Contents

1

This documentation is related to the LoggerSI Library v1.2.2

This library is supported on EcoStruxure™ Machine Expert HVAC 1.5.0 or greater versions.

Library description

LoggerSI library contains function blocks which allows to create 3 different types of data logger associated to the three main folders inside the library.

- Black Box
  - FB_BlackBox
- Cloud
  - CharlieJsonFormatGenerator
  - PubMMQTTmsgFromFile
  - GetTimeUTC
  - MqttIoTHub4Insight
- Datalogger
  - LogFile12iData
  - LogFileGenericData

There are still:

- Files utilities
  - AddStringToFile
  - CreationFilesList
  - ExportAllToUSB
  - FileNameCreation

- Structures:
  - strAsset
  - strDataAsset

    - strLogFileEvent
    - strLogFileGenStringData
    - strString
- LOGGERSI_CONSTS

Black Box

FB_BlackBox

This function block manages a fast data logging in RAM. uiSamplingPeriod defines a sampling period to store data.
Black box can manage a maximum of 512 sample for each data. Once black box is full, oldest data will be overide and xBlackBoxFull becomes TRUE.

When xSave becomes equal to TRUE, stored data are saved in the indicated CSV file.

xAppend=TRUE means that data will be saved in append mode to the existing CSV file if the file size is lower than udiAppendSwitchSize otherwise will be created a new CSV file.
If xAppend=FALSE, a new CSV file is created every time that a saving is required.

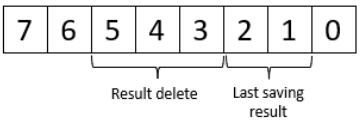
The name of the file is build taking in account: sFileDirectory, sPrefixFileName and sFileName (which must be indicated without extension).

The result will be a concatenation of these elements with an internal suffix that will be incremented at each save request or, in append mode, when the current file dimension is greater than udiMaxFileDim.

Example of File Name creation
sFileDirectory := 'SD/BBFolder/';
sPrefixFileName:= 'BB_';
sFileName := 'test';
Result: 'SD/BBFolder/BB_test_X.CSV' where X will be the incrementing suffix.

Runtime it's checked if there is still free space on the indicated path, if not the saving isn't allowed.
Thanks xDeleteFile and sFilePathToDelete, it's possible to delete a desired file when the saving process is not running.
When the function block is disabled, xEn=FALSE, usiStatus is resettled and the Black Box is cleaned.
sHeaderAddInfo will be printed on .csv file adding automatically # as first char.

usiStatus



Bit 0: logging
0: not logging
1: logging
Bit 1-2: Last saving result
00: do nothing
01: saving done
10: saving failed
Bit 3-4-5: Result delete
000: do nothing
001: delete done
010: delete failed
100: delete not available
Bit 6: error directory creation
Bit 7: Space full in the root indicated

Versions:
1.0.1
-Fix on sFileName dimension
1.0.2
-fix dimension of internal string
-increase dimension of data description
-internally printing of # as first char on sHeaderAddInfo row
1.0.3
- increased size of internal dimension
- added output information regarding the creation of a new file

Input:

Name	Type	Description
xEn	BOOL	Start/Stop data acquisition
uiSamplingPeriod	UINT	Sampling period to store data [s]
xSave	BOOL	Event to extract data from the black box and save them
sFileDirectory	STRING[32]	Directory where store the data extracted from the black box - default: 'SD/BBFolder/'
sPrefixFileName	STRING[32]	Prefix for the file name - default: ''
sFileName	STRING[32]	Your desired File name (without extension)
sHeaderAddInfo	STRING[100]	Row of usefull additional information.
xAppend	BOOL	TRUE=Append/Update to the existing .CSV file, FALSE=create at each xSave a new .CSV file
udiAppendSwitchSize	UDINT	In Append mode., if file's size >= [Kbytes], open a new file with the same name but suffix incremented - default: 2000 Kbytes

xDeleteFile	BOOL	Request to delete file
sFilePathToDelete	STRING[32]	File path to delete
uiData2log	UINT	Number of data to log
sDataDescription_1	STRING[32]	Label for data1 - default: 'AI1'
iData_1	INT	Data 1
usiDecimal_1	USINT	Format of data1: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_2	STRING[32]	Label for data2 - default: 'AI2'
iData_2	INT	Data 2
usiDecimal_2	USINT	Format of data2: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_3	STRING[32]	Label for data3 - default: 'AI3'
iData_3	INT	Data 3
usiDecimal_3	USINT	Format of data3: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_4	STRING[32]	Label for data4 - default: 'AI4'
iData_4	INT	Data 4
usiDecimal_4	USINT	Format of data4: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_5	STRING[32]	Label for data5 - default: 'AI5'
iData_5	INT	Data 5
usiDecimal_5	USINT	Format of data5: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_6	STRING[32]	Label for data6 - default: 'AI6'
iData_6	INT	Data 6
usiDecimal_6	USINT	Format of data6: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_7	STRING[32]	Label for data7 - default: 'AI7'
iData_7	INT	Data 7
usiDecimal_7	USINT	Format of data7: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_8	STRING[32]	Label for data8 - default: 'AI8'
iData_8	INT	Data 8
usiDecimal_8	USINT	Format of data8: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_9	STRING[32]	Label for data9 - default: 'AI9'
iData_9	INT	Data 9
usiDecimal_9	USINT	Format of data9: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_10	STRING[32]	Label for data10 - default: 'AI10'
iData_10	INT	Data 10

usiDecimal_10	USINT	Format of data10: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_11	STRING[32]	Label for data11 - default: 'AI11'
iData_11	INT	Data 11
usiDecimal_11	USINT	Format of data11: 0 = %.0f, 1 = %.1f, 2 = %.2f
sDataDescription_12	STRING[32]	Label for data12 - default: 'AI12'
iData_12	INT	Data 12
usiDecimal_12	USINT	Format of data12: 0 = %.0f, 1 = %.1f, 2 = %.2f

Output:

Name	Type	Description
usiStatus	USINT	Status information
sNameLastFile	STRING[75]	Name of the last file used to store data
udiLastFileSize	UDINT	Last File Dimension [Kbytes]
xBlackBoxFull	BOOL	TRUE = Black Box is full
xNewFileCreated	BOOL	Pulse to say that a new file has been created

Cloud

CharlieJsonFormatGenerator

This function block will create a string with Charlie json format.

It limits the output string to 255 characters.

If the maximum dimension is reached, xAlrmMaxDim becomes TRUE and sStringToWrite is truncated.

Example of Charlie json format:

```
{
    "metrics": {
        "assetName": "1",
        "Var1": <VAR_VALUE>,
        "Var1_timestamp": <TIMESTAMP_VALUE>,
        "Var2": <VAR_VALUE>,
        "Var2_timestamp": <TIMESTAMP_VALUE>
    }
}
```

Input:

Name	Type	Description
ptrasNameData	@strString	Pointer to array of data's name
ptraiDtataToSend	@INT	Pointer to array of data to send
iNbData	INT	Dimension of array

sAssetName	STRING[10]	Label for the asset field

Output:

Name	Type	Description
sStringGenerated	STRING[255]	Charlie as json format
xAlrmMaxDim	BOOL	Alarm maximum string dimension reached

GetTimeUTC

This function block returns the current UTC time with or without milliseconds.

Examples:
'2023-03-20T14:15:57Z'
'2023-03-20T14:15:57.523Z'

Input:

Name	Type	Description
xShow_ms	BOOL	Show milliseconds in the converted UTC time

Output:

Name	Type	Description
sUTCTime	STRING[24]	Current time with UTC format

MqttIoTHub4Insight

This function block creates a string with a json format compatible with the one required by Aveva Insight telemetry get from an Azure IoT Hub

Required json format:

```
{
    "TagName": "<<TagName>>",
    "EventTime": "<<time in UTC>>",
    "Quality": "<<Good/Questionable/Bad>>",
    "Value": "<<value>>"
 }
```

Versions:

V1.0.1

- Internal fix on FB dependence

Input:

Name	Type	Description
sTagName	STRING[64]	Tag Name
usiQuality	USINT	0=Good, 1=Questionable, else Bad - default:0=Good
rValue	REAL	Tag Value
sValueFormat	STRING[16]	String Format to be applied to rValue - Default '%.03f'

Output:

Name	Type	Description
sStringGenerated	STRING[255]	json message

PubMQTTmsgFromFile

This function block publishes on your desired MQTT broker some messages, which are read from a .txt file, available in sFilePath.

It manages two conditions:

- With MQTT Engine connection configured, uiEndpoint need to be coherent with the broker declared on Configuration
- Without MQTT Engine connection configured, uiEndpoint input must be greater than 100.

In case of MQTT Engine connection configure, before to publish the data read from file, the Engine connection is closed. Once data upload finished, xDone will become TRUE and the Engine connection will be open again if configured.

Input:

Name	Type	Description
xEnPublish	BOOL	Enables to publish MQTT messages from File
uiEndpointId	UINT	Broker Endpoint. If it's greater than 100, the endpoint is not configured on Configuration - default: 100
sBrokerAddress	STRING[255]	Broker Address
uiBrokerPort	UINT	Broker Port
sFilePath	STRING[255]	Path of file to read
sUsername	STRING[255]	Username
sPassword	STRING[255]	Password

sClientId	STRING[255]	Unique Client Id name
xEnTLS	BOOL	TRUE=TLS enabled, FALSE=TLS disabled
sCommonNameServer	STRING[255]	Required for TLS communication
udiTimeoutTcp	UDINT	Timeout for TCP operations [s] - default: 60s
udiTimeoutMqtt	UDINT	Timeout for MQTT operations [s] -default: 60s
uiWaitTimeToPublish	UINT	Time to wait before to publish a new data [0.1s] - default: 20
xDeleteFile	BOOL	Delete file once data are published - default: TRUE
sClientCert	STRING[255]	Optional client certificate filename
sClientKey	STRING[255]	Optional client private key associated to clientCert
sCACert	STRING[255]	Optional Root CA certificate filename to validate the broker

Output:

Name	Type	Description
usiState	USINT	Status information
xBusy	BOOL	Flag to say that MQTT connection is busy
xFileDeleted	BOOL	File successfully deleted
udiNbRecordFailed	UDINT	Number of records not published
udiNbRecordPublish	UDINT	Number of records published

usiState

Value	Name	Description
0	IDLE	Idle
1	FILE_OPENED	File opened
2	ENGINE_CLOSE	Engine connection closed
3	CONNECTION_OPENING	Connection Opening
4	CONNECTION_OPEN	Connection Opened
5	UPLOAD_IN_PROGRESS	Upload in progress
6	UPLOAD_ENDED	Upload Ended
7	UPLOAD_PARTIAL	Partial Upload
8	FILE_NOT_OPENED	File not opened
9	FILE_NOT_PRESENT	File not present

10	CONNECTION_FAILED	Connection Failed

DataLogger

LogFile12iData

This function block will save in a .csv file a defined set of integer data and events.

If xEn = TRUE, the sFilePath will be created. If it's already present., it will be used in append to record uiData2log and uiEvent2log elements.

Data recording period is specified thanks uiLogCycle. Events to log are stored on event.

If xDeletePreviousYear = TRUE, the file sFilePathLastYear will be automatically deleted.

If the number of data or events to log can be modified runtime and the current logging file will be reset.

sHeaderAddInfo will be printed on .csv file adding automatically # as first char.

If a clock alarm occurs, xAlarmClock=TRUE, then data acquisition is stopped. On previous file, a row is added with Status=3.

On the .csv file, the State indicates the logger state:

- 0=disables, it means that the logger has been manually stopped;
- 1=logger on, it means that the logger is running;
- 2=logger on after a reboot.
- 3=alarm clock occurs

Versions:
1.0.1
- Fix on sFileName dimension
- Added new 3 Alarms on bit 7,8,9
- Bug fix for the Alarms indications

1.0.2
- fix dimension of internal string
- increase dimension of data description
- internally printing of # as first char on sHeaderAddInfo row

1.0.3
- Added option for nul record at log start
- Increased buffer string size
- Add output information creation new file

1.0.4
- Add internal check about uiData2log and uiEvent2log

1.0.5
- Prevent file creation when MicroSD format is wrong

1.0.6
- - Add xAlarmClock management
- - Fix on generation not desired files
- - Fix on alarm when file not openend

Input:

Name	Type	Description
xEn	BOOL	Enables function block
sFilePath	STRING[99]	Path where create/modify your csv file (Ex. /SD/CSVFiles/LOG2022AUG.CSV)
sFileName	STRING[99]	Name of your csv file. (Ex. LOG2022AUG.CSV)
sFilePathLastYear	STRING[99]	Path where create/modify your csv file of the last year (Ex. /SD/CSVFiles/LOG2021AUG.CSV)
sHeaderAddInfo	STRING[100]	Row of usefull additional information.
uiLogCycle	UINT	Log period [s] - default: 60 sec
xDeletePreviousYear	BOOL	TRUE: delete file of previous year - default: TRUE
xRemoveFileReq	BOOL	Remove the file if present -default: FALSE
uiData2log	UINT	Number of data to log
sDataDescription_1	STRING[32]	Label for data1 - default: 'AI1'
iData_1	INT	Data 1
usiDecimal_1	USINT	Format of data1: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_2	STRING[32]	Label for data2 - default: 'AI2'
iData_2	INT	Data 2
usiDecimal_2	USINT	Format of data2: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_3	STRING[32]	Label for data3 - default: 'AI3'
iData_3	INT	Data 3
usiDecimal_3	USINT	Format of data3: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_4	STRING[32]	Label for data4 - default: 'AI4'
iData_4	INT	Data 4
usiDecimal_4	USINT	Format of data4: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_5	STRING[32]	Label for data5 - default: 'AI5'
iData_5	INT	Data 5
usiDecimal_5	USINT	Format of data5: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_6	STRING[32]	Label for data6 - default: 'AI6'
iData_6	INT	Data 6

usiDecimal_6	USINT	Format of data6: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_7	STRING[32]	Label for data7 - default: 'AI7'
iData_7	INT	Data 7
usiDecimal_7	USINT	Format of data7: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_8	STRING[32]	Label for data8 - default: 'AI8'
iData_8	INT	Data 8
usiDecimal_8	USINT	Format of data8: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_9	STRING[32]	Label for data9 - default: 'AI9'
iData_9	INT	Data 9
usiDecimal_9	USINT	Format of data9: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_10	STRING[32]	Label for data10 - default: 'AI10'
iData_10	INT	Data 10
usiDecimal_10	USINT	Format of data10: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_11	STRING[32]	Label for data11 - default: 'AI11'
iData_11	INT	Data 11
usiDecimal_11	USINT	Format of data11: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
sDataDescription_12	STRING[32]	Label for data12 - default: 'AI12'
iData_12	INT	Data 12
usiDecimal_12	USINT	Format of data12: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f
uiEvent2log	UINT	Number of events to log
usiEvent_1	USINT	Event 1
sEventDescription_1	STRING[12]	Label for Event1 - default: 'EV_1'
usiEvent_2	USINT	Event 2
sEventDescription_2	STRING[12]	Label for Event2 - default: 'EV_1'
usiEvent_3	USINT	Event 3
sEventDescription_3	STRING[12]	Label for Event3 - default: 'EV_1'
usiEvent_4	USINT	Event 4
sEventDescription_4	STRING[12]	Label for Event4 - default: 'EV_1'

usiEvent_5	USINT	Event 5
sEventDescription_5	STRING[12]	Label for Event5 - default: 'EV_1'
usiEvent_6	USINT	Event 6
sEventDescription_6	STRING[12]	Label for Event6 - default: 'EV_1'
usiEvent_7	USINT	Event 7
sEventDescription_7	STRING[12]	Label for Event7 - default: 'EV_1'
usiEvent_8	USINT	Event 8
sEventDescription_8	STRING[12]	Label for Event8 - default: 'EV_1'
usiEvent_9	USINT	Event 9
sEventDescription_9	STRING[12]	Label for Event9 - default: 'EV_1'
usiEvent_10	USINT	Event 10
sEventDescription_10	STRING[12]	Label for Event10 - default: 'EV_1'
usiEvent_11	USINT	Event 11
sEventDescription_11	STRING[12]	Label for Event11 - default: 'EV_1'
usiEvent_12	USINT	Event 12
sEventDescription_12	STRING[12]	Label for Event12 - default: 'EV_1'
xLimitFile2Mb	BOOL	If TRUE, file dimension is limited to 2Mb
xAddNullRecordAtStart	BOOL	Add a Null a record at log start
xAlarmClock	BOOL	TRUE=an alarm clock occurs; stop acquisition

Output

Name	Type	Description
udiFileDimension	UDINT	File Dimension [byte]
xRecordSaved	BOOL	New record saved
udiRecordCounter	UDINT	Record Counter
uiAlarmID	UINT	AlarmID
xAlarm	BOOL	Logger Alarm
xNewFileCreated	BOOL	Pulse to say that a new file has been created

Alarm ID:

Bit	Value	Description
1	2	Alarm active when the log error number becomes greater than 1000. When active, it disables the log

2	4	Alarm file not opened or writing not valid
4	16	Alarm uiData2log greater than the size of the data array
5	32	Alarm uiEvent2log greater than the size of the event array
6	64	Alarm file longer than 2Mb
7	128	Invalid task
8	256	Mismatch between sFileName and sFilePath
9	512	Error during directory creation
10	1024	Alarm clock active
11	2048	Alarm file not opened

LogFileGenericData

This function block will save in a .csv file a defined set of data andevents.

If xEn = TRUE, the sFilePath will be created. If it's already present.,it will be used in append to record uiData2log and uiEvent2log elements.

Data recording period is specified thanks uiLogCycle. Events to log are stored on event.

If xDeletePreviousYear = TRUE, the file sFilePathLastYear will be automatically deleted.

The number of data or events to log can be modified runtime and the current logging file will be reset.

sHeaderAddInfo will be printed on .csv file adding automatically # as first char.

If a clock alarm occurs, xAlarmClock=TRUE, then data acquisition is stopped. On previous file, a row is added with Status=3.

On the .csv file, the State indicates the logger state:

- 0=disables, it means that the logger has been manually stopped;
- 1=logger on, it means that the logger is running;
- 2=logger on after a reboot.
- 3=Alarm clock occurs

Versions:
1.0.1
- Fix on sFileName dimension
- Added new 3 Alarms on bit 7,8,9
- Bug fix for the Alarms indications

1.0.2
- Added Option for NULL record at log start and boot
- Increased size of internal buffer

1.0.3

- Added output information regarding the creation of a new file

1.0.4

- Add internal check about uiData2log and uiEvent2log

1.0.5

- Prevent file creation when MicroSD format is wrong

1.0.6

- - Add xAlarmClock management
- - Fix on generation not desired files
- - Fix on alarm when file not openend

Input

Name	Type	Description
xEn	BOOL	Enables function block
sFilePath	STRING[99]	Path where create/modify your csv file (Ex. /SD/CSVFiles/LOG2022AUG.CSV)
sFileName	STRING[99]	Name of your csv file. (Ex. LOG2022AUG.CSV)
sFilePathLastYear	STRING[99]	Path where create/modify your csv file of the last year (Ex. /SD/CSVFiles/LOG2021AUG.CSV)
sHeaderAddInfo	STRING[100]	Row of usefull additional information.
uiLogCycle	UINT	Log period [s] - default: 60 sec
xDeletePreviousYear	BOOL	TRUE: delete file of previous year - default: TRUE
xRemoveFileReq	BOOL	Remove the file if present -default: FALSE
uiData2log	UINT	Number of data to log
ptrLoggedData	@strLogFileGenSignedData	Pointer to vector of signed data to be logged
uiEvent2log	UINT	Number of events to log - Max: 128
ptrLoggedEvent	@strLogFileEvent	Pointer to vector of event to be logged
xLimitFileSize	BOOL	If TRUE, file size is limited to udiMaxFileSize bytes
udiMaxFileSize	UDINT	Max log file size [bytes] - Default 2000000
xAddNullRecordAtStart	BOOL	Add a Null a record at log start
xAlarmClock	BOOL	TRUE=an alarm clock occurs; stop acquisition

Output:

Name	Type	Description
udiFileDimension	UDINT	File Size [byte]

xRecordSaved	BOOL	New record saved
udiRecordCounter	UDINT	Record Counter
uiAlarmID	UINT	AlarmID
xAlarm	BOOL	Logger Alarm
xNewFileCreated	BOOL	Pulse to say that a new file has been created

Alarm ID:

Bit	Value	Description
1	2	Alarm active when the log error number becomes greater than 1000. When active, it disables the log
2	4	Alarm file not opened or writing not valid
4	16	Alarm uiData2log greater than the size of the data array
5	32	Alarm uiEvent2log greater than the size of the event array
6	64	Alarm file longer than 2Mb
7	128	Invalid task
8	256	Mismatch between sFileName and sFilePath
9	512	Error during directory creation
10	1024	Pointer to event array not valid
11	2048	Pointer to data array not valid
12	4096	Alarm clock active
13	8192	File not opened

Files utilities

AddStringToFile

When xEn = TRUE, this function block adds on request a string to a file opened in append mode.

It returns the number of records saved on your file.

Input/Output:

Name	Type	Description
xAddStringToFile	BOOL	Request to add string to file

Input:

Name	Type	Description
xEn	BOOL	Enables the function block
sBckFilePath	STRING[255]	Path where to save your file
sStringToWrite	STRING[255]	String to add on the file

Output:

Name	Type	Description
udiRecordCounter	UDINT	Number of records saved on the file

CreationFilesList

The goal of this function block is to save the path of files with a specific extension and available in sPath.

This list of paths is stored in sPathFilesList.

In order to have always an updated list of files' path, when xGenerateList passes from FALSE to TRUE, if the file sPathLoggedFilesList already exists it is deleted and re-created.

If there are no error during the creation, xFileCreated will be TRUE. In case of issue, you can refer to LastError output for debug.

Versions
1.0.1
- Fix on string length to write on the file

1.0.2
- Fix on buffer length
- Code optimization

Input/Output

Name	Type	Description
xGenerateList	BOOL	TRUE: generate 'FileList.txt'

Input:

Name	Type	Description
sPath	STRING[255]	Path where search all the files with a particular extension – default: 'SD/CSVFiles/'
sExtension	STRING[10]	Extension file to search – default:'*.CSV'

sPathFilesList	STRING[255]	Path where save your Files List – default: 'SD/FilesList.txt'

Output:

Name	Type	Description
xFileCreated	BOOL	If TRUE, file is created
LastError	FS_ERROR	Last error of File System function

ExportAllToUSB

The goal of this function block is to export to USB a copy of files with a specific extension, sExtension, present in sPathSource.

V1.0.1

- Bug fix on internal string length

Input/Output:

Name	Type	Description
xExportAll	BOOL	TRUE: export all files on USB

Input:

Name	Type	Description
sPathSource	STRING[255]	Path where search all the files with a particular extension - default: '/SD/CSVFiles/'
sExtension	STRING[10]	Extension file to search - default: '*.CSV'

Output:

Name	Type	Description
xDone	BOOL	Procedure Ended
xFileExported	BOOL	If TRUE, one file is exported
LastError	FS_ERROR	Last error of File System function
uiNbFilesCopied	UINT	Number of copied files

FileNameCreation

This function block will generate a file's NAME starting from the inputs indicated.

It'll create also a folder, if required and it doesn't exist. In case of invalid calling which causes a folder's creation failed, xDone becomes FALSE.

Some possibility for sFileDirectory are:

- '/SD/FolderName' or '/SD'
- '/SHARED_FS/FolderName' or '/SHARED_FS', NAND shared with FTP and Web site
- '/FolderName' or '/', private NAND
- '/USB/FolderName' or '/USB' </p>

The name format will be: sPrefix + yyyymmmdd + sExtension.

Example:

sFileDirectory := '/SD/CSVFiles';

sPrefix := 'LOG';

sExtension := '.CSV';

The result will be:

sFilePath := '/SD/CSVFiles/LOG2022NOV20.CSV';

sFileName := 'LOG2022NOV20.CSV'; .

Versions:
1.0.1
- Prevent file creation when MicroSD format is wrong

Input:

Name	Type	Description
sFileDirectory	STRING[32]	Directory where save the csv file - default: '/SD/CSVFiles'
sPrefix	STRING[32]	File prefix - default: 'LOG'
sExtension	STRING[32]	File extension - default: '.CSV'
DateAndTime	DATE_AND_TIME	Date and Time indication
xIsFilePerMonth	BOOL	TRUE = 1 file per month FALSE= 1 file per day

Output:

Name	Type	Description
xDone	BOOL	Result of directory creation
sFilePath	STRING[100]	Path of your csv file
sFileName	STRING[32]	Name of your csv file

sFilePathLastYear	STRING[32]	Path of the csv file of last year
sFileNameLastYear	STRING[32]	Name of the csv file of last year

Structures

strAsset

strAsset structure is used to populate the Charlie JSON format used to send data to Cloud.

strDataAsset structure is mandatory with strAsset.

Charlie JSON format:

```
{
    "metrics": {
        "assetName": "1",
        "Var1": <VAR_VALUE>,
        "Var1_timestamp": <TIMESTAMP_VALUE>,
        "Var2": <VAR_VALUE>,
        "Var2_timestamp": <TIMESTAMP_VALUE>
    }
}
```

Name	Type	Description
xEn	BOOL	Enables this asset
sAssetName	STRING[10]	Asset Name
udiPeriodAsset	UDINT	Period time on when send your message [ms]
iNbData	INT	Number of variable present on your message. Must be <=(NBVAR_ASSET_MINUS1+1) - default: 8
DataInfo	strDataAsset	Data information: variables' name and value, send on variation, threshold
uiInhibitTime	UINT	Inhibit time [s]
xAlarmCharlieDim	BOOL	Alarm maximum string dimension reached

strDataAsset

This structure is used to set information related to asset's variables.

It requires to define a constant NBVAR_ASSET_MINUS_1 which will be the number of variables to use with the current asset.

Name	Type	Description
asVarName	ARRAY [0..NBVAR_ASSET_MINUS1] of strString	Variable name

aiVarValue	ARRAY [0..NBVAR_ASSET_MINUS1] of INT	Variable value
axSendOnVar	ARRAY [0..NBVAR_ASSET_MINUS1] of BOOL	Send on variation
auiThreshold	ARRAY [0..NBVAR_ASSET_MINUS1] of UINT	Threshold

strLogFileEvent

Single element to be logged by LogFileGenericData

Name	Type	Description
ptrusiEvent	DWORD	Logged Event (address of a BOOL or 8 bit var)
sDescription	STRING[12]	Description

strLogFileGenSignedData

Single Element to be logged by LogFileGenericData

Name	Type	Description
ptrValue	DWORD	Pointer to Logged Value
usiSizeWord	USINT	Number of words 1-2 otherwise is assumed 4
usiDecimal	USINT	Format of data1: 0 = %.0f, 1 = %.1f, 2 = %.2f, 3=%.3f. In case of integer values data is also divided by 10^usiDecimal
xIsFloat	BOOL	Used if usiSizeWord is 2 or 4. If true value is assumed as a REAL otherwise as a signed integer
sDescription	STRING[12]	Description
xIsDataOutOfRange	BOOL	If TRUE value is not written

strString

This structure is used to create array of STRING

Name	Type	Description
sString	STRING[255]	Desired string name

LOGGERSI_CONSTS

Name	Type	Description
NBVAR_ASSET_MINUS1	INT	Maximum number of variables on a MQTT asset minus 1 – Init value:7

Safety Information

A Appendix

Before You Begin

General. The products specified in this document have been tested under actual service conditions. Of course, your specific application requirements may be different from those assumed for this and any related examples described herein. In that case, you will have to adapt the information provided in this and other related documents to your particular needs. To do so, you will need to consult the specific product documentation of the hardware and/or software components that you may add or substitute for any examples specified in this documentation. Pay particular attention and conform to any safety information, different electrical requirements and normative standards that would apply to your adaptation.

© 2020 Schneider Electric. All rights reserved.

WARNING: REGULATORY INCOMPATIBILITY

Be sure that all equipment applied and systems designed comply with all applicable local, regional and national regulations and standards. Failure to follow these instructions can result in death, serious injury, or equipment damage.

Note. Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved. Failure to observe this information can result in injury or equipment damage.

The use and application of the information contained herein require expertise in the design and programming of automated control systems. Only the user or integrator can be aware of all the conditions and factors present during installation and setup, operation, and maintenance of the machine or process, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used. When selecting automation and control equipment, and any other related equipment or software, for a particular application, the user or integrator must also consider any applicable local, regional or national standards and/or regulations.

Some of the major software functions and/or hardware components used in the proposed architectures and examples described in this document cannot be substituted without significantly compromising the performance of your application. Further, any such substitutions or alterations may completely invalidate any proposed architectures, descriptions, examples, instructions, wiring diagrams and/or compatibilities between the various hardware components and software functions specified herein and in related documentation. You must be aware of the consequences of any modifications, additions or substitutions.

A residual risk, as defined by EN/ISO 12100-1, Article 5, will remain if:

- it is necessary to modify the recommended logic and if the added or modified components are not properly integrated in the control circuit;
- you do not follow the required standards applicable to the operation of the machine, or if the adjustments to and the maintenance of the machine are not properly made (it is essential to strictly follow the prescribed machine maintenance schedule);
- the devices connected to any safety outputs do not have mechanically-linked contacts.

CAUTION: EQUIPMENT INCOMPATIBILITY

Read and thoroughly understand all device and software documentation before attempting any component substitutions or other changes related to the application examples provided in the document. Failure to follow these instructions can result in injury, or equipment damage.

Start-Up and Test

Before using electrical control and automation equipment after design and installation, the application and associated functional safety system must be subjected to a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such testing be made and that enough time is allowed to perform complete and satisfactory testing.

CAUTION: EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in injury, or equipment damage.

Verify that the completed system, including the functional safety system, is free from all short circuits and grounds, except those grounds installed according to local regulations. If high-potential voltage testing is necessary, follow the recommendations in equipment documentation to help prevent injury or equipment damage.

Operation and Adjustments

Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly installed and operated.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the hands and other parts of the body are free to enter the pinch points or other hazardous areas where serious injury can occur. Software products alone cannot protect an operator from injury. For this reason, the software cannot be substituted for or take the place of point-of-operation protection.

WARNING: UNGUARDED MACHINERY CAN CAUSE SERIOUS INJURY

- Do not use this software and related automation equipment on equipment which does not have point-of -operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

Note. Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the examples and implementations suggested herein. It is sometimes possible to adjust the equipment incorrectly and this produce unsatisfactory or unsafe operation. Always use the manufacturer instructions as a guide to functional adjustments. Personnel who have access to these adjustments must be familiar with the equipment manufacturer instructions and the machinery used with the electrical equipment.

Only those operational adjustments actually required by the machine operator should be accessible to the operator. Access to other controls should be restricted to help prevent unauthorized changes in operating characteristics.

Limitation of liability. The user retains full responsibility for testing the hardware and software solution where LoggerSI library is used.