LibDatalogger
User Manual

MU214611 Rev. B

May 3, 2021

www.altus.com.br

No part of this document may be copied or reproduced in any form without the prior written consent of Altus Sistemas de Automação S.A. who reserves the right to carry out alterations without prior advice.

According to current legislation in Brazil, the Consumer Defense Code, we are giving the following information to clients who use our products, regarding personal safety and premises.

The industrial automation equipment, manufactured by Altus, is strong and reliable due to the stringent quality control it is subjected to. However, any electronic industrial control equipment (programmable controllers, numerical commands, etc.) can damage machines or processes controlled by them when there are defective components and/or when a programming or installation error occurs. This can even put human lives at risk. The user should consider the possible consequences of the defects and should provide additional external installations for safety reasons. This concern is higher when in initial commissioning and testing.

The equipment manufactured by Altus does not directly expose the environment to hazards, since they do not issue any kind of pollutant during their use. However, concerning the disposal of equipment, it is important to point out that built-in electronics may contain materials which are harmful to nature when improperly discarded. Therefore, it is recommended that whenever discarding this type of product, it should be forwarded to recycling plants, which guarantee proper waste management.

It is essential to read and understand the product documentation, such as manuals and technical characteristics before its installation or use. The examples and figures presented in this document are solely for illustrative purposes. Due to possible upgrades and improvements that the products may present, Altus assumes no responsibility for the use of these examples and figures in real applications. They should only be used to assist user trainings and improve experience with the products and their features.

Altus warrants its equipment as described in General Conditions of Supply, attached to the commercial proposals.

Altus guarantees that their equipment works in accordance with the clear instructions contained in their manuals and/or technical characteristics, not guaranteeing the success of any particular type of application of the equipment.

Altus does not acknowledge any other guarantee, directly or implied, mainly when end customers are dealing with third-party suppliers. The requests for additional information about the supply, equipment features and/or any other Altus services must be made in writing form. Altus is not responsible for supplying information about its equipment without formal request. These products can use EtherCAT® technology (www.ethercat.org).

COPYRIGHTS

Nexto, MasterTool, Grano and WebPLC are the registered trademarks of Altus Sistemas de Automação S.A.

Windows, Windows NT and Windows Vista are registered trademarks of Microsoft Corporation.

OPEN SOURCE SOFTWARE NOTICE

To obtain the source code under GPL, LGPL, MPL and other open source licenses, that is contained in this product, please contact opensource@altus.com.br. In addition to the source code, all referred license terms, warranty disclaimers and copyright notices may be disclosed under request.

Contents

1. Introduction . . . 1
   1.1. Technical Support . . . 1
   1.2. Warning Messages Used in this Manual . . . 1
2. Datalogger Working Principles . . . 2
   2.1. Non-Volatile Storage Media . . . 2
   2.2. Multiple Instances of FB . . . 2
   2.3. Folders . . . 2
   2.4. CSV Files . . . 3
   2.5. Copying CSV Files to a Computer . . . 4
   2.6. Selection of Values to be Logged . . . 4
   2.7. Types of Values to be Logged . . . 4
   2.8. Continuous and Trigger Modes . . . 5
      2.8.1. Continuous Mode . . . 5
      2.8.2. Trigger Mode . . . 6
         2.8.2.1. Special Cases of Trigger Mode with Reduced Number of Records . . . 8
         2.8.2.2. Special Case with Missed Triggers . . . 10
   2.9. Behavior of REC_INTERVAL and Sample Time of FB DATALOGGER . . . 10
   2.10. Circular and Non-Circular Storage of Files . . . 10
      2.10.1. Non-Circular Behavior . . . 11
      2.10.2. Circular Behavior . . . 11
   2.11. Types of CSV Delimiters . . . 11
   2.12. Formatting TIME Column in Excel® and Similar Spreadsheets . . . 11
   2.13. Usage of FB in Redundant CPUs . . . 11
   2.14. Timing Diagrams of Boolean Outputs (DONE, BUSY, ERROR) . . . 12
      2.14.1. Errors Detected in Input Parameters at Rising Edge of ENABLE . . . 12
      2.14.2. Errors Detected Later when ENABLE is TRUE . . . 12
      2.14.3. Behavior of Continuous Mode Saving Files all the Time . . . 13
      2.14.4. Behavior of Continuous Mode Saving Files until Number of Files is Reached . . . 13
      2.14.5. Normal Behavior in Trigger Mode . . . 14
      2.14.6. Behavior in Trigger Mode with File Being Captured during Falling Edge of ENABLE . . . 15
      2.14.7. Behavior in Trigger Mode when Maximum Number of Files is Reached . . . 15
3. Input and Output Parameters of Function Block DATALOGGER . . . 16
   3.1. Input Parameters . . . 16
      3.1.1. ENABLE - Enable execution of FB . . . 16
      3.1.2. FILE_NAME - Prefix for Naming Folders and CSV Files . . . 17
      3.1.3. STORAGE - Type of Non-Volatile Storage . . . 17
      3.1.4. FILE_RECORDS - Maximum Number of Records in a CSV File in Continuous Mode . . . 17
      3.1.5. NUMBER_OF_FILES - Maximum Number of CSV Files per Folder . . . 17
      3.1.6. MODE - Datalogger Mode . . . 18

3.1.7. CIRCULAR - Circular Storage of Files . . . 18
3.1.8. TRIGGER_BEFORE - Number of Records before Trigger . . . 18
3.1.9. TRIGGER_AFTER - Number of Records after Trigger . . . 18
3.1.10. TRIGGER - Trigger Event . . . 18
3.1.11. REC_INTERVAL - Time Interval between Records of CSV Files . . . 18
3.1.12. COMMA_DELIMITER - Types of CSV Delimiters . . . 19
3.1.13. FILE_HEADER - Definition of Final Part of File Header . . . 19
3.1.14. INPUT001 ... INPUT256 - Value Inputs for Logging . . . 19
3.2. Output Parameters . . . 19
3.2.1. DONE - Completed Record Writing . . . 19
3.2.2. BUSY - Busy Writing Records . . . 20
3.2.3. ERROR - Error or Status Condition Active . . . 20
3.2.4. STATUS - Error or Status Description . . . 20
3.2.5. CURRENT_RECORDS - Number of Records Saved . . . 20
3.2.6. MISSED_TRIGGERS - Number of Missed Trigger Events . . . 20
4. Application Example . . . 21

1. Introduction

This manual describes library *LibDatalogger*. This library makes available one function block (*DATALOGGER*) and some related enumerations, as shown in following figure. The FB *DATALOGGER* allows the user to record data in an organized way in CSV files stored in non-volatile memory, so they can be visualized thereafter in spreadsheets applications like Excel®.

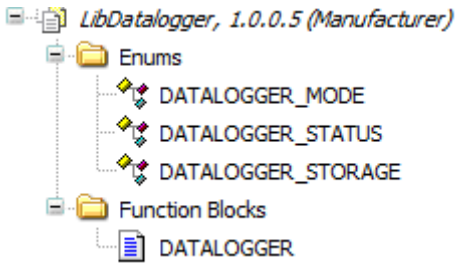


Figure 1: Components of library LibDatalogger (version 1.0.0.5 and newest)

ATTENTION:
This manual is applicable for version 1.0.0.5 or newest of library *LibDatalogger*. Older versions of this library do not contain all the features described in this manual.
This library is compatible with Nexto Series PLCs and XTORM Series RTUs.

The following chapters describe this library in detail:

- Chapter Datalogger Working Principles describes how datalogger works and associated concepts;
- Chapter Input and Output Parameters of Function Block DATALOGGER describes in detail all the input and output parameters of DATALOGGER FB;
- Chapter Application Example provides an example of applications using the FB DATALOGGER.

1.1. Technical Support

For Altus Technical Support contact in São Leopoldo, RS, call +55 51 3589-9500. For further information regarding the Altus Technical Support existent on other places, see https://www.altus.com.br/en/ or send an email to altus@altus.com.br.

If the equipment is already installed, you must have the following information at the moment of support requesting:

- The model from the used equipments and the installed system configuration
- The product serial number
- The equipment revision and the executive software version, written on the tag fixed on the product's side
- CPU operation mode information, acquired through MasterTool IEC XE
- The application software content, acquired through MasterTool IEC XE
- Used programmer version

1.2. Warning Messages Used in this Manual

In this manual, the warning messages will be presented in the following formats and meanings:

DANGER:
Reports potential hazard that, if not detected, may be harmful to people, materials, environment and production.

CAUTION:
Reports configuration, application or installation details that must be taken into consideration to avoid any instance that may cause system failure and consequent impact.

ATTENTION:
Identifies configuration, application and installation details aimed at achieving maximum operational performance of the system.

2. Datalogger Working Principles

This chapter describes the working principles and important concepts related to FB DATALOGGER.

The following figure shows the interfaces (input and output parameters) of this FB. Some of these parameters will be referenced in this chapter while describing the working principles. A more detailed description of input and output parameters is given in chapter Input and Output Parameters of Function Block DATALOGGER.

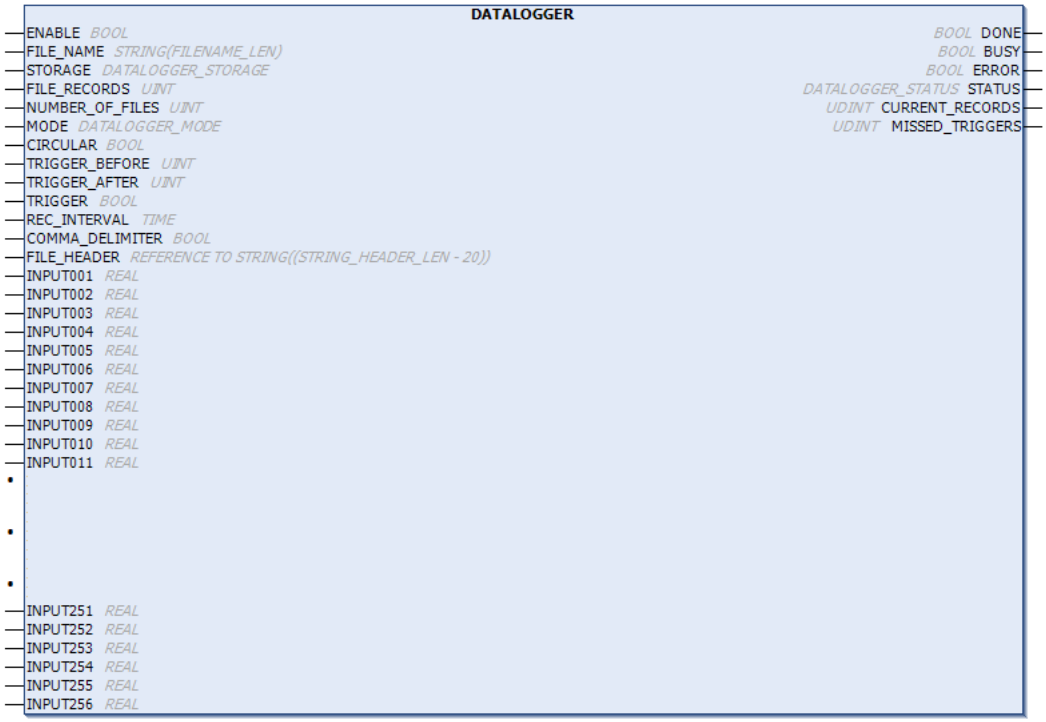


Figure 2: Parameters of FB DATALOGGER

2.1. Non-Volatile Storage Media

The CSV files created by the FB can be stored in the following types of non-volatile media:

- Internal FLASH memory (all models of Nexto PLCs and XTORM RTUs);
- Memory cards (some models of Nexto PLCs and XTORM RTUs);
- USB flash drive (some models of Nexto PLCs).

The storage media is selected by input parameter STORAGE of FB.

ATTENTION:
External storage devices (memory cards and USB flash drives) must be formatted with FAT32 file system. See user manual of related Nexto and XTORM CPUs for information about compatible external storage devices and formatting instructions.

2.2. Multiple Instances of FB

Multiple instances of FB DATALOGGER can be created.

2.3. Folders

The CSV files created by FB DATALOGGER instances are stored in folders. Folders and files can be accessed using Mastertool programmer, clicking on *Device* in the devices tree, and then selecting tab *Files*.

The folder where CSV files are stored are created below the following folders:

- **InternalMemory/Datalogger:** if internal FLASH memory is selected as storage media;
- **MemoryCard/Datalogger:** if an external memory card is selected as storage media;
- **Mass_Storage/Datalogger:** if an external USB flash drive is selected as storage media.

The folder *Datalogger*, as well folders below folder *Datalogger*, are automatically created by the FB.

The following figure shows an example with 4 folders below folder *Datalogger*:

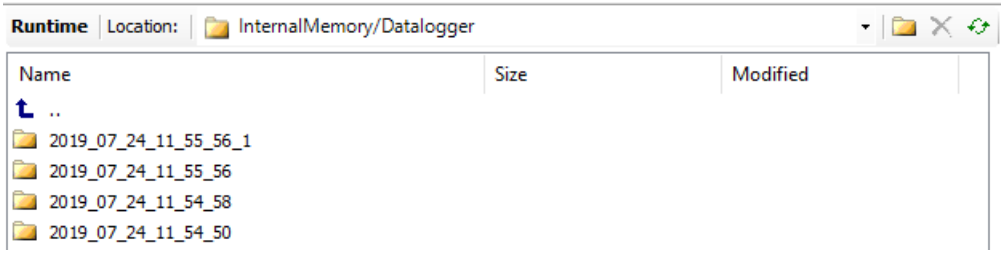


Figure 3: Example of folders for Datalogger files

The name of a folder for storing CSV files is formed using UTC date and time when the folder was created, followed by an optional suffix, with the following format:

<year>_<month>_<day>_<hour>_<minute>_<second>[_<optional suffix>]

A new folder is created by an instance of FB DATALOGGER each time a rising edge is detected in input parameter ENABLE. This behavior is useful for restarting the datalogger function, especially when input parameter CIRCULAR is FALSE (see more details in section Circular and Non-Circular Storage of Files).

In previous figure, the first folder has an optional suffix (_1) so that it can be differentiated from the second folder. An optional suffix is generated to avoid duplicated folder names, in the following situations:

- When two or more instances of FB DATALOGGER rise their inputs ENABLE at the same time. In this case, the first instance that creates the folder has no suffix, and the next instances will have suffixes _1, _2, etc.
- If clock is adjusted backwards, there is a chance of a new rising edge of input ENABLE occurring at same time when a folder was created in the past. In this case, the new folder is also created with an option suffix.

2.4. CSV Files

CSV files store the data collected by a FB instance. These files are stored in folders, as described in the previous section.

The following figure shows an example where 5 CSV files are stored inside a folder.

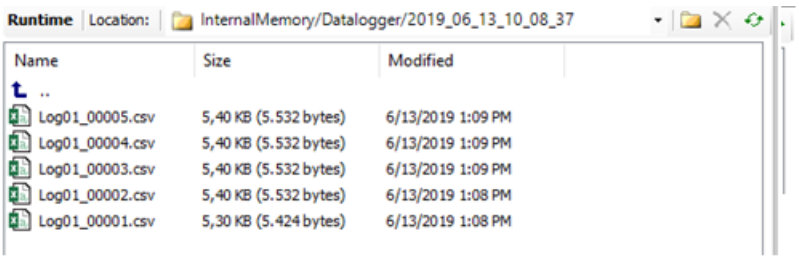


Figure 4: Example of CSV files of Datalogger

The names of files are formed according the following rule:

<FILE_NAME>_<sequential number>.csv

- <FILE_NAME> corresponds to input parameter FILE_NAME of FB. Each instance of FB must have a different value for this input parameter;
- <sequential number>: this number is generated sequentially, and starts with 00001 when a rising edge is detected in input parameter ENABLE.

ATTENTION:
Different instances of FB DATALOGGER must have different values for input parameter FILE_NAME. If this recommendation is not followed, the user may get corrupted CSV files if these instances are storing their CSV files in the same folder (this may occur when rising edge of input ENABLE is detected at the same time by the different instances - see section Folders.

The maximum number of CSV files created by an instance of FB DATALOGGER inside a folder is defined by input parameter NUMBER_OF_FILES.

The moment when a CSV file is closed and another is created with the next sequential number, is controlled by several input parameters (MODE, FILE_RECORDS, TRIGGER, TRIGGER_BEFORE, TRIGGER_AFTER, REC_INTERVAL). See section Continuous and Trigger Modes for more details about this (this section also describe the formats of CSV files in continuous and trigger modes).

The sequential number of a CSV file varies between 00001 and NUMBER_OF_FILES if input parameter CIRCULAR is FALSE. However, if input parameter CIRCULAR is TRUE, sequential numbers are not limited to range 00001 to NUMBER_OF_FILES. See section Circular and Non-Circular Storage of Files for more details about this.

2.5. Copying CSV Files to a Computer

The files stored in the PLC non-volatile storage can be copied to a computer using the "«" button in the Mastertool screen shown in the following figure.

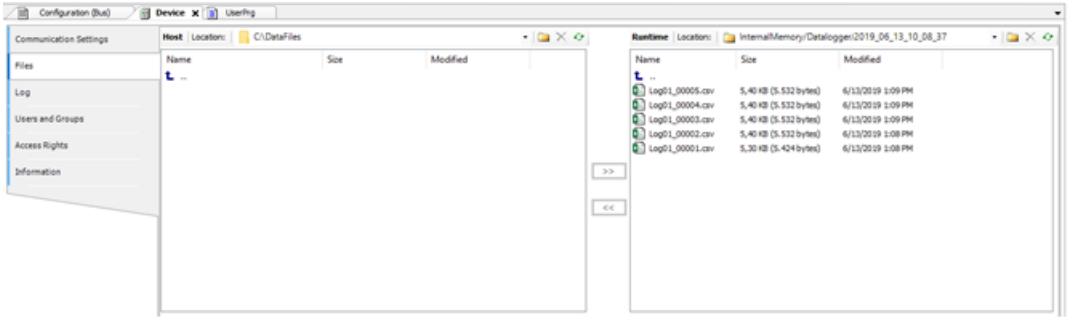


Figure 5: Copying CSV files from PLC to a computer

2.6. Selection of Values to be Logged

Each instance of FB DATALOGGER allow to log up to 256 input values, corresponding to input parameters INPUT001 to INPUT256.

The number of logged inputs may vary between 1 and 256, and this is defined by input parameter FILE_HEADER. This input parameter defines the final part of the header (first line) of each CSV file, so it is a string. For instance consider:

FILE_HEADER = 'V1;V2;V3' (with input parameter COMMA_DELIMITER = FALSE)

or

FILE_HEADER = 'V1,V2,V3' (with input parameter COMMA_DELIMITER = TRUE)

In this example, the CSV files will store three values under columns named V1, V2 and V3, and these values correspond respectively to input parameters INPUT001, INPUT002 and INPUT003.

For information about input parameter COMMA_DELIMITER, see section Types of CSV Delimiters.

2.7. Types of Values to be Logged

The input parameters INPUT001 to INPUT256 have type REAL. However, it is also possible to log other types of values. The following table shows all the types of values that can be directly assigned to an input parameters INPUT001 to INPUT256.

Value type	Lower limit	Upper limit
SINT	-128	127
USINT / BYTE	0	255
INT	-32768	32767

Value type	Lower limit	Upper limit
UINT / WORD	0	65535
DINT	-8388608	8388608
UDINT / DWORD	0	8388608
LINT	-8388608	8388608
ULINT / LWORD	0	8388608
REAL	-	-
LREAL	$\pm 10^{-44.85}$	$\pm 10^{+38.53}$

Table 1: Supported data types and limits

ATTENTION:
The limits ±8388608 were defined so that conversions from types DINT, UDINT, LINT and ULINT to type REAL occur without losing precision. Outside these limits, some loss of precision may occur.
When converting from type LREAL to type REAL, the absolute value of the LREAL must be in the range $10^{-44.85}$ to $10^{+38.53}$ (the special value 0 is also supported). Furthermore, even if a LREAL is inside this range, some loss of precision may occur when converting to REAL.

Using some auxiliary code, it is possible to log other types not mentioned in previous table. For instance, if the user wants to log a type BOOL variable, he can convert it to a SINT auxiliary variable, and then assign the SINT variable to an input variable (INPUT001 ... INPUT256).

```
IF BOOL_VARIABLE THEN
  SINT_VARIABLE := 1;
ELSE
  SINT_VARIABLE := 0;
```

2.8. Continuous and Trigger Modes

The input parameter MODE allows to select between two datalogger modes:

- MODE = CONTINUOUS_MODE
- MODE = TRIGGER_MODE

The following subsections describe each of these two modes.

2.8.1. Continuous Mode

In continuous mode, the selected input values are recorded cyclically. The cycle for each record is defined by input parameter REC_INTERVAL.

When the number of records in a CSV file reaches input parameter FILE_RECORDS, this CSV file is closed, and a new CSV file is opened with the next sequential number (see section CSV Files).

The following figure shows an example of CSV file (Log01_00002.csv) shown by Excel® application, with the following input parameters:

- MODE = CONTINUOUS_MODE
- REC_INTERVAL = T#100ms
- FILE_RECORDS = 100
- FILE_HEADER = 'V1,V2,V3'
- FILE_NAME = 'Log01'
- COMMA_DELIMITER = TRUE (see section Types of CSV Delimiters for details about this parameter).

	A	B	C	D	E	F
1	RECORD	DATE	TIME	V1	V2	V3
2	101	14/06/2019	09:20:21.914	827.6989	1655.398	2483.09
3	102	14/06/2019	09:20:22.014	828.81	1657.62	2486.423
4	103	14/06/2019	09:20:22.114	829.921	1659.842	2489.756
5	104	14/06/2019	09:20:22.214	831.032	1662.064	2493.089
6	105	14/06/2019	09:20:22.314	832.143	1664.286	2496.422
7	106	14/06/2019	09:20:22.414	833.2541	1666.508	2499.755
8	107	14/06/2019	09:20:22.514	834.3651	1668.73	2503.088
9	108	14/06/2019	09:20:22.614	835.4761	1670.952	2506.421
10	109	14/06/2019	09:20:22.714	836.5871	1673.174	2509.754
11	110	14/06/2019	09:20:22.814	837.6981	1675.396	2513.087
12	111	14/06/2019	09:20:22.914	838.8092	1677.618	2516.42
13	112	14/06/2019	09:20:23.014	839.9202	1679.84	2519.753
14	113	14/06/2019	09:20:23.114	841.0312	1682.062	2523.086
15	114	14/06/2019	09:20:23.214	842.1422	1684.284	2526.419
16	115	14/06/2019	09:20:23.314	843.2532	1686.507	2529.752
17	116	14/06/2019	09:20:23.414	844.3643	1688.729	2533.085
18	117	14/06/2019	09:20:23.514	845.4753	1690.951	2536.418
19	118	14/06/2019	09:20:23.614	846.5863	1693.173	2539.751
			• • •			
88	187	14/06/2019	09:20:30.514	923.2469	1846.494	2769.729
89	188	14/06/2019	09:20:30.614	924.3579	1848.716	2773.062
90	189	14/06/2019	09:20:30.714	925.4689	1850.938	2776.395
91	190	14/06/2019	09:20:30.814	926.58	1853.16	2779.728
92	191	14/06/2019	09:20:30.914	927.691	1855.382	2783.061
93	192	14/06/2019	09:20:31.014	928.802	1857.604	2786.394
94	193	14/06/2019	09:20:31.114	929.913	1859.826	2789.727
95	194	14/06/2019	09:20:31.214	931.024	1862.048	2793.06
96	195	14/06/2019	09:20:31.314	932.1351	1864.27	2796.393
97	196	14/06/2019	09:20:31.414	933.2461	1866.492	2799.726
98	197	14/06/2019	09:20:31.514	934.3571	1868.714	2803.059
99	198	14/06/2019	09:20:31.614	935.4681	1870.936	2806.392
100	199	14/06/2019	09:20:31.714	936.5792	1873.158	2809.725
101	200	14/06/2019	09:20:31.814	937.6902	1875.38	2813.058

Figure 6: Example of CSV file in continuous mode (Log01_00002.csv)

First column (RECORD) always shows the record number. Note that the example has 100 records, numbered from 101 to 200. This happened because:

- FILE_RECORDS = 100
- Sequential number of the file is 00002 (Log01_00002.csv). If the sequential number was 00001, then records would have numbers varying from 1 to 100. But with sequential number 00002, records have numbers varying from 101 to 200.

Second column (DATE) shows the date when the record was captured.

Third column (TIME) shows the time of day when the record was captured (from hour to millisecond).

ATTENTION:
DATE and TIME columns refer to UTC time instead of local time.

From fourth column on, the values corresponding to value input parameters are shown. In this case, three columns with values exist, because input parameter FILE_HEADER = 'V1,V2,V3'.

In continuous mode, each CSV file always contain a number of records equal to FILE_RECORDS, and the time difference between consecutive records is REC_INTERVAL (see difference between column TIME of consecutive records).

2.8.2. Trigger Mode

In trigger mode, the selected input values are recorded according to a trigger condition, defined by input parameter TRIGGER.

In each rising edge of TRIGGER input, one CSV file is normally created, and the number of records of this file depends on input parameters TRIGGER_BEFORE and TRIGGER_AFTER. The number of records of this file is normally equal to:

TRIGGER_BEFORE + 1 + TRIGGER_AFTER

The following figure shows an example of CSV file (Log01_00002.csv) shown by Excel® application, with the following input parameters:

- MODE = TRIGGER_MODE
- REC_INTERVAL = T#1s
- TRIGGER_BEFORE = 5
- TRIGGER_AFTER = 10
- FILE_HEADER = 'V1,V2,V3'
- FILE_NAME = 'Log01'
- COMMA_DELIMITER = TRUE (see section Types of CSV Delimiters for details about this parameter).

In this example, it is also important to mention that FB instance is being called in a cyclic task with period equal to 100 ms.

	A	B	C	D	E	F
1	RECORD	DATE	TIME	V1	V2	V3
2	-5	14/06/2019	09:11:49.542	989.9083	1979.817	2969.709
3	-4	14/06/2019	09:11:50.542	1001.019	2002.037	3003.039
4	-3	14/06/2019	09:11:51.542	1012.129	2024.258	3036.369
5	-2	14/06/2019	09:11:52.542	1023.239	2046.478	3069.699
6	-1	14/06/2019	09:11:53.542	1034.349	2068.697	3103.029
7	0	14/06/2019	09:11:54.042	1039.903	2079.807	3119.694
8	1	14/06/2019	09:11:54.542	1045.458	2090.917	3136.36
9	2	14/06/2019	09:11:55.542	1056.568	2113.136	3169.69
10	3	14/06/2019	09:11:56.542	1067.678	2135.355	3203.02
11	4	14/06/2019	09:11:57.542	1078.787	2157.574	3236.35
12	5	14/06/2019	09:11:58.542	1089.897	2179.794	3269.68
13	6	14/06/2019	09:11:59.542	1101.006	2202.013	3303.01
14	7	14/06/2019	09:12:00.542	1112.116	2224.232	3336.34
15	8	14/06/2019	09:12:01.542	1123.226	2246.451	3369.67
16	9	14/06/2019	09:12:02.542	1134.335	2268.671	3403
17	10	14/06/2019	09:12:03.542	1145.445	2290.89	3436.33

Figure 7: Example of CSV file in trigger mode (Log01_00002.csv)

First column (RECORD) shows the record number. Note that the example has 16 records, numbered from -5 to 10. This happened because TRIGGER_BEFORE = 5 and TRIGGER_AFTER = 10.

During trigger mode, the values are still sampled with period equal to REC_INTERVAL, but they normally are not stored in the CSV file.

When the rising edge of input parameter TRIGGER is detected:

- The TRIGGER_BEFORE records previously sampled with REC_INTERVAL period are saved in CSV file. In the example of previous figure, these correspond to records -5 to -1.
- The values sampled exactly when rising edge of input parameter TRIGGER was detected are saved in CSV file as record 0. Note that sampling of TRIGGER input is done with sample time of the FB instance, that is, 100 ms in this example.
- The next TRIGGER_AFTER records sampled will be saved in CSV file. In the example of previous figure, this corresponds to records 1 to 10.

It is important to note that time difference between records -1 e 1 is equal to REC_INTERVAL (see column TIME of these records). The TIME of record 0, corresponding to rising edge of TRIGGER input, is in between TIMEs of records -1 and 1, because it is sampled faster (in this example, the sample time of FB instance is 100 ms).

Second column (DATE) shows the date when the record was captured.

Third column (TIME) shows the time of day when the record was captured (from hour to millisecond).

ATTENTION:
DATE and TIME columns refer to UTC time instead of local time.

From fourth column on, the values corresponding to value input parameters are shown. In this case, three columns with values exist, because input parameter FILE_HEADER = 'V1,V2,V3'.

2.8.2.1. Special Cases of Trigger Mode with Reduced Number of Records

In some special circumstances, the number of records saved in a CSV file may be less than normal (TRIGGER_BEFORE + 1 + TRIGGER_AFTER).

These circumstances are described through some examples that follow in this subsection. In all these examples, assume the same input parameters established for previous example in this subsection:

- MODE = TRIGGER_MODE
- REC_INTERVAL = T#1s
- TRIGGER_BEFORE = 5
- TRIGGER_AFTER = 10
- FILE_HEADER = 'V1,V2,V3'
- FILE_NAME = 'Log01'
- COMMA_DELIMITER = TRUE

The first special circumstance occurs when the rising edge of TRIGGER is detected less than TRIGGER_BEFORE * REC_INTERVAL after first falling edge of BUSY that occurs after rising edge of ENABLE. In the example illustrated by the following figure, rising edge of TRIGGER was detected 2.5 seconds after first falling edge of BUSY that occurred after rising edge of ENABLE. Note that less than TRIGGER_BEFORE records were saved before record 0 (records -5, -4 and -3 are missing).

	A	B	C	D	E	F
1	RECORD	DATE	TIME	V1	V2	V3
2	-2	22/07/2019	17:22:52.202	918.8028	1837.606	2756.397
3	-1	22/07/2019	17:22:53.202	929.913	1859.826	2789.727
4	0	22/07/2019	17:22:53.902	937.6902	1875.38	2813.058
5	1	22/07/2019	17:22:54.202	941.0233	1882.047	2823.057
6	2	22/07/2019	17:22:55.202	952.1335	1904.267	2856.387
7	3	22/07/2019	17:22:56.202	963.2437	1926.487	2889.717
8	4	22/07/2019	17:22:57.202	974.3539	1948.708	2923.047
9	5	22/07/2019	17:22:58.202	985.4642	1970.928	2956.377
10	6	22/07/2019	17:22:59.202	996.5744	1993.149	2989.707
11	7	22/07/2019	17:23:00.202	1007.685	2015.369	3023.037
12	8	22/07/2019	17:23:01.202	1018.795	2037.59	3056.367
13	9	22/07/2019	17:23:02.202	1029.905	2059.81	3089.697
14	10	22/07/2019	17:23:03.202	1041.014	2082.029	3123.027

Figure 8: Example with little time between rising edges of ENABLE and TRIGGER

The second special circumstance occurs when a rising edge of TRIGGER is detected less than TRIGGER_BEFORE * REC_INTERVAL after falling edge of BUSY that occurs after a previous rising edge of TRIGGER. In the example illustrated by the two following figures, the second rising edge of TRIGGER occurs 2.5 seconds after the falling edge of BUSY that occurs after the first rising edge of TRIGGER:

- The first figure is associated with the first rising edge of TRIGGER, and the CSV file is complete (TRIGGER_BEFORE + TRIGGER_AFTER + 1 records).
- The second figure is associated with the second rising edge of TRIGGER, and the CSV file is not complete (less than TRIGGER_BEFORE + TRIGGER_AFTER + 1 records). Note that the 3 first records that are missing in the second CSV file should be equal to the 3 last records in the first CSV file. So, in this example, log information was not missed, because there is a continuity between the two CSV files.

	A	B	C	D	E	F
1	RECORD	DATE	TIME	V1	V2	V3
2	-5	22/07/2019	17:56:33.002	23373.91	46747.81	70099.42
3	-4	22/07/2019	17:56:34.002	23385.02	46770.04	70132.78
4	-3	22/07/2019	17:56:35.002	23396.13	46792.26	70166.14
5	-2	22/07/2019	17:56:36.002	23407.25	46814.49	70199.5
6	-1	22/07/2019	17:56:37.002	23418.36	46836.72	70232.86
7	0	22/07/2019	17:56:37.602	23425.03	46850.05	70252.88
8	1	22/07/2019	17:56:38.002	23429.47	46858.94	70266.22
9	2	22/07/2019	17:56:39.002	23440.59	46881.17	70299.58
10	3	22/07/2019	17:56:40.002	23451.7	46903.4	70332.94
11	4	22/07/2019	17:56:41.002	23462.81	46925.62	70366.3
12	5	22/07/2019	17:56:42.002	23473.93	46947.85	70399.66
13	6	22/07/2019	17:56:43.002	23485.04	46970.08	70433.02
14	7	22/07/2019	17:56:44.002	23496.15	46992.3	70466.38
15	8	22/07/2019	17:56:45.002	23507.26	47014.53	70499.74
16	9	22/07/2019	17:56:46.002	23518.38	47036.76	70533.1
17	10	22/07/2019	17:56:47.002	23529.49	47058.98	70566.46

Figure 9: First CSV file with little time between rising edges of two consecutive TRIGGERs

	A	B	C	D	E	F
1	RECORD	DATE	TIME	V1	V2	V3
2	-2	22/07/2019	17:56:48.002	23540.6	47081.21	70599.81
3	-1	22/07/2019	17:56:49.002	23551.72	47103.44	70633.17
4	0	22/07/2019	17:56:49.702	23559.5	47118.99	70656.53
5	1	22/07/2019	17:56:50.002	23562.83	47125.66	70666.53
6	2	22/07/2019	17:56:51.002	23573.94	47147.89	70699.89
7	3	22/07/2019	17:56:52.002	23585.06	47170.12	70733.25
8	4	22/07/2019	17:56:53.002	23596.17	47192.34	70766.61
9	5	22/07/2019	17:56:54.002	23607.28	47214.57	70799.97
10	6	22/07/2019	17:56:55.002	23618.4	47236.8	70833.33
11	7	22/07/2019	17:56:56.002	23629.51	47259.02	70866.69
12	8	22/07/2019	17:56:57.002	23640.62	47281.25	70900.05
13	9	22/07/2019	17:56:58.002	23651.74	47303.48	70933.41
14	10	22/07/2019	17:56:59.002	23662.85	47325.7	70966.77

Figure 10: Second CSV file with little time between rising edges of two consecutive TRIGGERs

In the two previous circumstances, some CSV files were saved with reduced number of records because the corresponding rising edge of TRIGGER occurred inside a "trigger before zone" (see definition of "trigger before zone" in timing diagram of Figure 17).

The third and last special circumstance occurs when a falling edge of ENABLE is detected less than TRIGGER_AFTER * REC_INTERVAL after rising edge of TRIGGER. In the example of following figure, falling edge of ENABLE was detected 6.5 seconds after rising edge of TRIGGER. Therefore, less than TRIGGER_AFTER records were saved after record 0 (records 8, 9 and 10 are missing).

	A	B	C	D	E	F
1	RECORD	DATE	TIME	V1	V2	V3
2	-5	22/07/2019	18:28:46.702	44847.95	89695.89	134614.8
3	-4	22/07/2019	18:28:47.702	44859.04	89718.08	134648.1
4	-3	22/07/2019	18:28:48.702	44870.14	89740.27	134681.4
5	-2	22/07/2019	18:28:49.702	44881.23	89762.45	134714.7
6	-1	22/07/2019	18:28:50.702	44892.32	89784.64	134747.9
7	0	22/07/2019	18:28:51.402	44900.09	89800.17	134771.2
8	1	22/07/2019	18:28:51.702	44903.42	89806.83	134781.2
9	2	22/07/2019	18:28:52.702	44914.51	89829.02	134814.5
10	3	22/07/2019	18:28:53.702	44925.6	89851.2	134847.8
11	4	22/07/2019	18:28:54.702	44936.7	89873.39	134881.1
12	5	22/07/2019	18:28:55.702	44947.79	89895.58	134914.4
13	6	22/07/2019	18:28:56.702	44958.89	89917.77	134947.6
14	7	22/07/2019	18:28:57.702	44969.98	89939.95	134980.9

Figure 11: Example with little time between rising edge of TRIGGER and falling edge of ENABLE

In the previous circumstance, the CSV file was saved with reduced number of records because the corresponding rising edge of TRIGGER occurred inside a "trigger after zone" (see definition of "trigger after zone" in timing diagram of Figure 17).

2.8.2.2. Special Case with Missed Triggers

If a rising edge of TRIGGER occurs while output BUSY is TRUE, this trigger event is missed, and output MISSED_TRIGGER is incremented (see definition of "missed trigger zone" in timing diagram of Figure 17).

In trigger mode, the BUSY output goes up together with the TRIGGER input, and its maximum expected duration is:

*TRIGGER_AFTER * REC_INTERVAL*

2.9. Behavior of REC_INTERVAL and Sample Time of FB DATALOGGER

First of all, it is important to emphasize that FB DATALOGGER must be called cyclically, that is, in a cyclic task. The period or interval of this cyclic task is defined as the sample time (SampleTime) of FB DATALOGGER.

This subsection defines the behavior of input parameter REC_INTERVAL and its relation with SampleTime.

No matter which is the selected MODE (continuous mode or trigger mode), the value inputs (INPUT001 ... INPUT256) are sampled first time in the rising edge of input ENABLE. After this, these variables are normally sampled with a period equal to REC_INTERVAL. This normal behavior happens when REC_INTERVAL is a multiple of SampleTime (for instance, if SampleTime = 100 ms and REC_INTERVAL = 1 second).

The user normally should make REC_INTERVAL input be a multiple of SampleTime. If this recommendation is not followed, the value inputs are sampled with the next higher REC_INTERVAL that is multiple of SampleTime. For instance:

- If SampleTime = 100 ms and REC_INTERVAL = 250 ms, then the sample period will be 300 ms;
- If SampleTime = 100 ms and REC_INTERVAL = 50 ms, then the sample period will be 100 ms;

Some remarks must be made about the trigger mode (see section Trigger Mode and observe Figure 7):

- In trigger mode not all samples are saved in CSV files, because a trigger condition must exist in TRIGGER input for saving the samples;
- In trigger mode record number 0 is not sampled with with period equal to REC_INTERVAL. Instead, the special record 0 is sampled in the rising edge of TRIGGER input using period SampleTime.

2.10. Circular and Non-Circular Storage of Files

The input parameter NUMBER_OF_FILES defines the maximum number of CSV files that should be stored bellow a folder allocated by an FB instance. The input parameter CIRCULAR defines how the FB DATALOGGER manages this quantity of files.

After rising edge of ENABLE input, a folder is created, and the first CSV file will be created with sequential number 00001.

2.10.1. Non-Circular Behavior

When input parameter CIRCULAR is FALSE, the FB instance will stop to create CSV files after the number of saved CSV files gets equal to NUMBER_OF_FILES.

For instance, if NUMBER_OF_FILES is 5, then sequential number of files will vary between 00001 and 00005, and after this the FB will stop to create new CSV files. If the FB is demanded to create a sixth CSV file, the ERROR output will be activated, and the STATUS output will be set with value ERROR_NUMBER_OF_FILES_REACHED.

The only way to create new files with this FB instance is to generate a new rising edge in ENABLE input. In this case, a new folder will be created for the new CSV files, and the old CSV files will stay in the old folder.

2.10.2. Circular Behavior

When input parameter CIRCULAR is TRUE, the FB instance continues to create CSV files after number of saved CSV files gets equal to NUMBER_OF_FILES. While NUMBER_OF_FILES files are already completed and saved, an additional CSV file is prepared. After this additional file is completed it is saved and the oldest CSV file is deleted, to keep the number of CSV files equal to NUMBER_OF_FILES.

The sequential number is incremented for each new saved file, so it can get bigger than NUMBER_OF_FILES. When sequential number reaches 65535, next file will be created with sequential number 0, and after this it starts to be incremented again towards 65535.

When input ENABLE of FB DATALOGGER switchs to FALSE, the remaining number of files in the folder can be NUMBER_OF_FILES or NUMBER_OF_FILES + 1. The number of files will be NUMBER_OF_FILES + 1 if the newest file was not complete when input ENABLE switched to FALSE. A non-complete file means that its number of records is:

- Smaller than FILE_RECORDS, in continuous mode;
- Smaller than TRIGGER_BEFORE + TRIGGER_AFTER + 1, in trigger mode.

2.11. Types of CSV Delimiters

The input parameter COMMA_DELIMITER configures some delimiters used by FB DATALOGGER, according to regional settings.

COMMA_DELIMITER should be TRUE in countries like US. In this case:

- A comma (,) is used as delimiter between columns of CSV file, in the header (see input parameter FILE_HEADER) and also in the following lines of CSV file.
- A dot (.) is used as a decimal point:
  - In column TIME, between seconds and milliseconds;
  - In the columns used for recording values.

COMMA_DELIMITER should be FALSE in countries like Brazil. In this case:

- A semicolon (;) is used as delimiter between columns of CSV file, in the header (see input parameter FILE_HEADER) and also in the following lines of CSV file.
- A comma (,) is used as a decimal point:
  - In column TIME, between seconds and milliseconds;
  - In the columns used for recording values.

2.12. Formatting TIME Column in Excel® and Similar Spreadsheets

After opening the CSV file, for proper exhibition of TIME column, its format must be customized, according the following rules:

If COMMA_DELIMITER = FALSE, use format "hh:mm:ss.000"

If COMMA_DELIMITER = TRUE, use format "hh:mm:ss,000"

2.13. Usage of FB in Redundant CPUs

In redundant CPUs, FB DATALOGGER must be instanced in a non-redundant memory area.

Furthermore, it must be always executed in both PLCs (active and non-active). Therefore, it must be called in a POU like NonSkippedPrg (see user manual of Nexto and XTORM CPUs for details about redundancy).

2.14. Timing Diagrams of Boolean Outputs (DONE, BUSY, ERROR)

This section shows some time diagrams for describing the behavior of bollean outputs DONE, BUSY and ERROR, complementing the description of these outputs given in chapter Input and Output Parameters of Function Block DATALOGGER.

The timing diagrams also include bollean inputs ENABLE and TRIGGER, because the behavior of aforementioned outputs are related to these inputs.

2.14.1. Errors Detected in Input Parameters at Rising Edge of ENABLE

The following timing diagram shows the behavior of DONE, BUSY and ERROR when there are errors in input parameters at rising edge of input ENABLE.

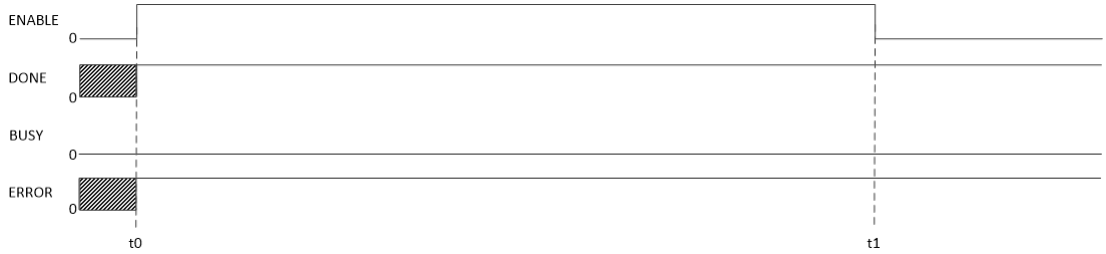


Figure 12: Timing with errors in input parameters

Before time t0 (time when ENABLE input goes up):

- DONE can be TRUE of FALSE. It will be FALSE if input ENABLE has never been TRUE since power-on or since off-line download of application to PLC; otherwise it will be TRUE;
- BUSY is FALSE;
- ERROR can be TRUE of FALSE, depending on error detection in previous situation FB has executed with ENABLE equal TRUE.

At time t0 the ENABLE input goes up, and at this time:

- DONE and ERROR will change to TRUE, because of error detection in input parameters;
- BUSY remains FALSE.
-

At time t1 the ENABLE input goes down, and nothing changes in outputs DONE, BUSY and ERROR.

2.14.2. Errors Detected Later when ENABLE is TRUE

The following timing diagram shows the behavior of DONE, BUSY and ERROR when some special errors are detected later (not during rising edge of ENABLE), but while ENABLE is still TRUE. For instance, such a special error occurs if the external memory card or USB storage is dismounted while FB DATALOGGER is trying to save CSV files on it.

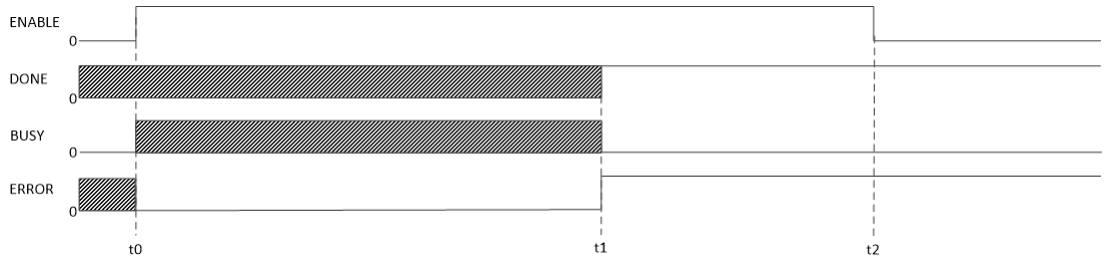


Figure 13: Timing with special errors detected later

Before time t0 (rising edge of input ENABLE), the behavior of DONE, BUSY and ERROR is the same described in section Errors Detected in Input Parameters at Rising Edge of ENABLE.

A special error is detected at time t1. Between t0 and t1:

- Behavior of DONE and BUSY depends on several factors that will better explained in the following subsections;
- ERROR is FALSE.

At time t1:

- DONE and ERROR will change to TRUE, because of special error detection;
- BUSY changes to FALSE.

At time t2 the ENABLE input goes down, and nothing changes in outputs DONE, BUSY and ERROR.

2.14.3. Behavior of Continuous Mode Saving Files all the Time

The following diagram applies when input MODE is CONTINUOUS_MODE, and FB DATALLOGER keeps saving CSV files during all the time when input ENABLE is TRUE.

This happens when input CIRCULAR is TRUE, or when ENABLE input is switched off before maximum number of files were saved (input NUMBER_OF_FILES). Furthermore, errors must not be detected while input ENABLE is TRUE.

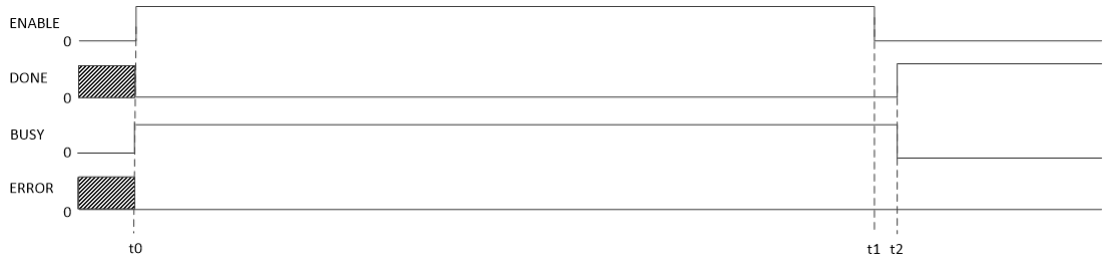


Figure 14: Timing with continuous mode saving files all the time

Before time t0 (rising edge of input ENABLE), the behavior of DONE, BUSY and ERROR is the same described in section Errors Detected in Input Parameters at Rising Edge of ENABLE.

At time t0:

- DONE changes to FALSE;
- BUSY changes to TRUE;
- ERROR changes to FALSE.

At time t1 (falling edge of ENABLE), nothing happens immediately with outputs DONE, BUSY and ERROR.

At time t2 (small delay after t1):

- DONE changes to TRUE;
- BUSY changes to FALSE;
- ERROR stays in FALSE.

The small delay between t1 and t2 is the time necessary for completing saving the CSV files in the selected storage media.

2.14.4. Behavior of Continuous Mode Saving Files until Number of Files is Reached

The following diagram applies when input MODE is CONTINUOUS_MODE, and FB DATALLOGER stops saving CSV files before input ENABLE changes to FALSE.

This happens when input CIRCULAR is FALSE and the maximum number of files (input NUMBER_OF_FILES) is reached before falling edge of input ENABLE.

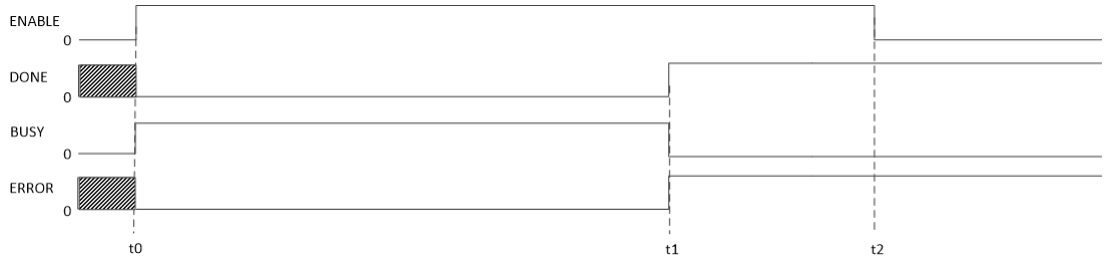


Figure 15: Timing with continuous mode stopping to save files

Before time t0 (rising edge of input ENABLE), the behavior of DONE, BUSY and ERROR is the same described in section Errors Detected in Input Parameters at Rising Edge of ENABLE.

At time t0:

- DONE changes to FALSE;
- BUSY changes to TRUE;
- ERROR changes to FALSE.

At time t1, when the number of files save reaches input NUMBER_OF_FILES:

- DONE changes to TRUE;
- BUSY changes to FALSE;
- ERROR changes to TRUE (the error specified in output STATUS is ERROR_NUMBER_OF_FILES_REACHED).

At time t2 (falling edge of ENABLE), nothing happens with outputs DONE, BUSY and ERROR.

2.14.5. Normal Behavior in Trigger Mode

The following timing diagram shows the normal behavior of DONE, BUSY and ERROR in trigger mode. A normal behavior is understood as:

- No errors are found while ENABLE is TRUE;
- No triggers are missed (rising edge of TRIGGER is out of "missed trigger zones");
- No CSV files are saved with less than "TRIGGER_BEFORE + TRIGGER_AFTER + 1" records (rising edge of trigger is out of "trigger before zones" and "trigger after zones");

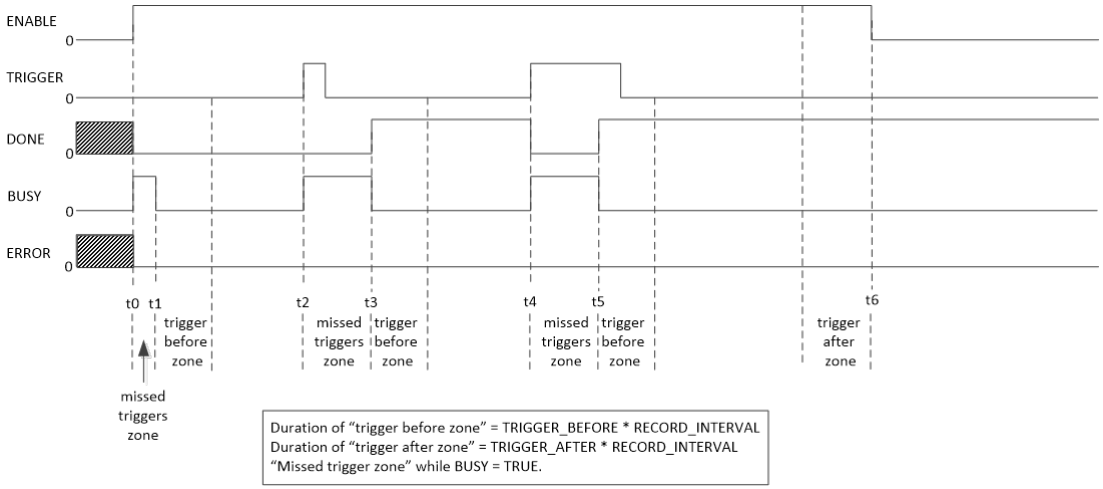


Figure 16: Timing with normal trigger mode

A "missed trigger zone" occurs while BUSY output is TRUE. If a rising edge of TRIGGER is detected inside a "missed trigger zone", this trigger will be missed, that is, a corresponding CSV file will not be saved, and the output MISSED_TRIGGERS will be incremented.

In trigger mode, the BUSY output is TRUE in the following circumstances:

- In the small time interval between t0 and t1. This interval is needed for creating the directory and queue necessary for saving the CSV files in the storage media;
- During some time after rising edge of a non-missed TRIGGER input. This interval is needed for capturing the remaining TRIGGER_AFTER records and saving the CSV files after this. The duration of this interval is approximately equal to TRIGGER_AFTER * REC_INTERVAL.

A "trigger before zone" starts in the falling edge of BUSY, and its duration is equal to TRIGGER_BEFORE * REC_INTERVAL. If a rising edge of TRIGGER is detected inside a "trigger before zone", the corresponding CSV file may have less than TRIGGER_BEFORE records with negative values in column RECORD (records captured before rising edge of TRIGGER).

A "trigger after zone" ends in the falling edge of ENABLE, and its duration is equal to TRIGGER_AFTER * REC_INTERVAL. If a rising edge of TRIGGER is detected inside a "trigger after zone", the corresponding CSV file may have less than TRIGGER_AFTER records with positive values in column RECORD (records captured after rising edge of TRIGGER).

2.14.6. Behavior in Trigger Mode with File Being Captured during Falling Edge of ENABLE

The following timing diagram is almost equal to that shown in previous section, except that an additional rising edge of TRIGGER occurs during the "trigger after zone".

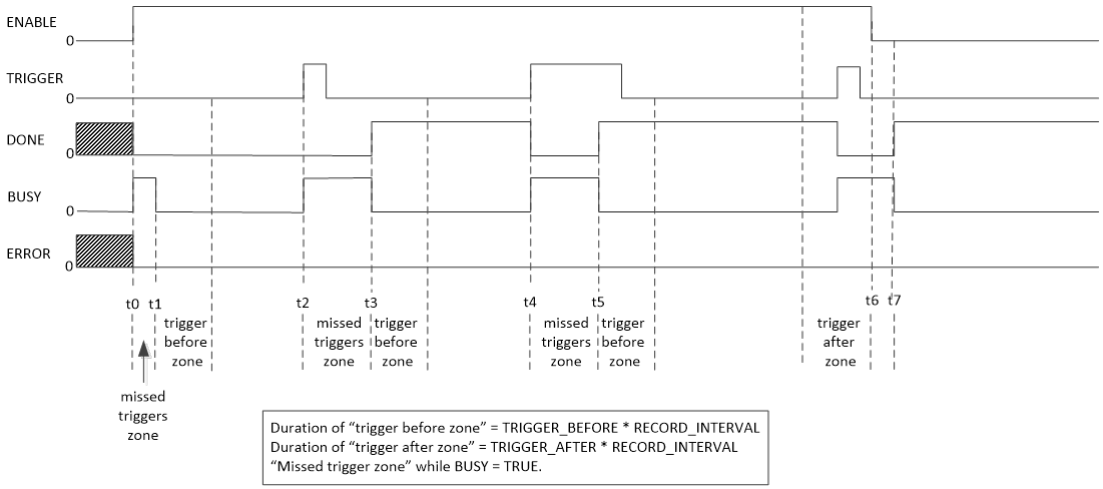


Figure 17: Timing capturing file during falling edge of ENABLE

In this case, at time t7, DONE goes high and BUSY goes low. Difference between t7 and t6 is a small time needed for saving the last file in the storage media.

2.14.7. Behavior in Trigger Mode when Maximum Number of Files is Reached

If input parameter CIRCULAR is FALSE, the following timing may be observed if the maximum number of files (input parameter NUMBER_OF_FILES) is reached.

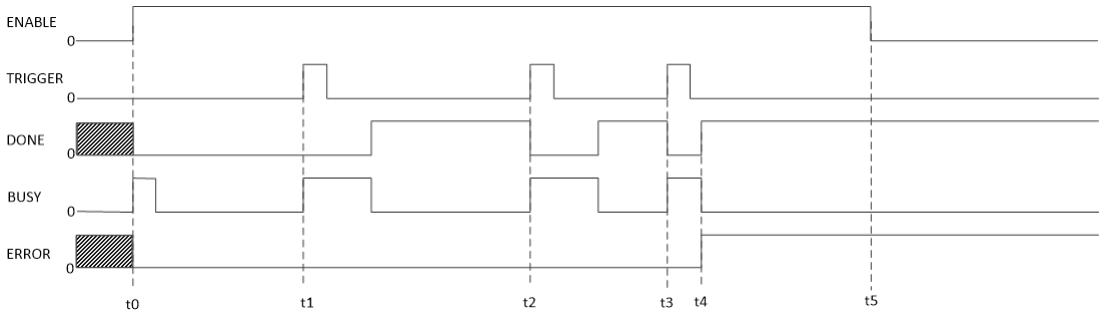


Figure 18: Timing capturing file after maximum number of files reached

In this example, assume that input parameter NUMBER_OF_FILES is equal 2. These two files are associated with rising edges of TRIGGER at times t1 and t2. The third rising edge of TRIGGER at time t3 does not create a third CSV file. Instead, at t4 (small delay after t3) the ERROR output goes HIGH, and STATUS output gets equal to ERROR_NUMBER_OF_FILES_REACHED.

3. Input and Output Parameters of Function Block DATALOGGER

Input and output parameters of FB DATALOGGER are shown in the following figure.

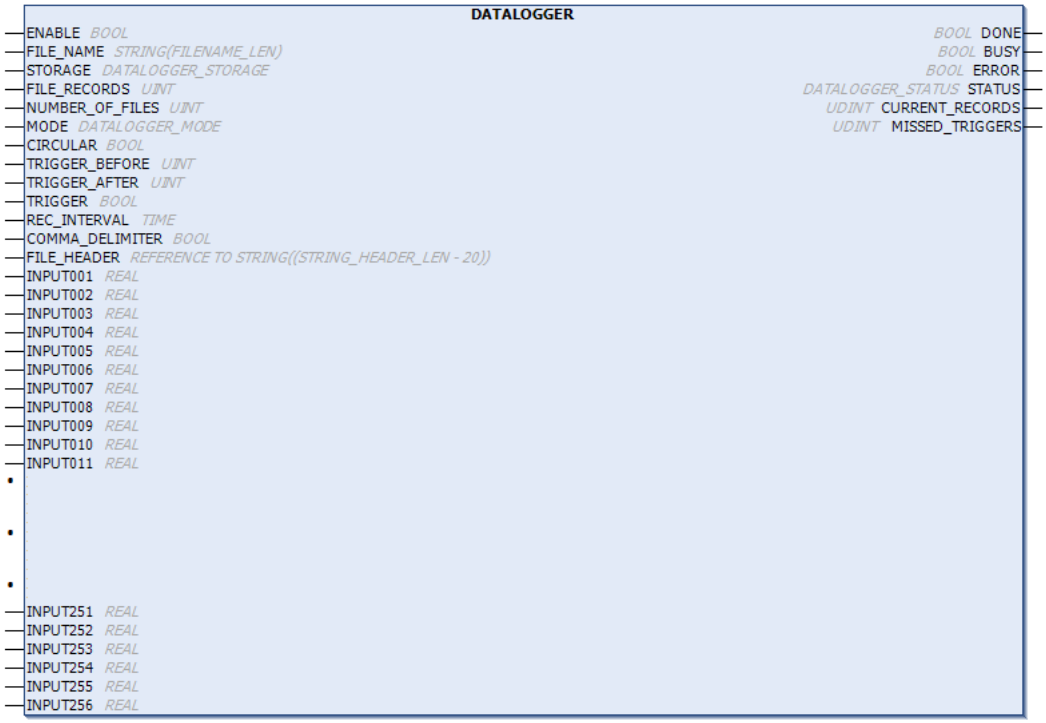


Figure 19: Parameters of FB DATALOGGER

3.1. Input Parameters

The following subsections describe the input parameters.

> **ATTENTION:**
> **For all inputs, a default value is indicated. If the input can keep the default value all the time, this input does not need to be explicitly passed as an input parameter when calling the FB instance (in graphical languages, its pin does not need to be connected).**

3.1.1. ENABLE - Enable execution of FB

- Type: BOOL
- Default value: FALSE

This input must be TRUE for enabling the datalogger function.

In the rising edge of this input, a new folder is created for storing the CSV files. For details, see section Folders in chapter Datalogger Working Principles.

Also in the rising edge of this input, the following input parameters are checked for consistence, causing error messages in output parameter STATUS if some inconsistence is detected:

- FILE_NAME
- STORAGE
- FILE_RECORDS
- NUMBER_OF_FILES
- MODE
- CIRCULAR
- TRIGGER_BEFORE

- TRIGGER_AFTER
- REC_INTERVAL
- COMMA_DELIMITER
- FILE_HEADER

> **ATTENTION:**
> **The previously mentioned input parameters are ignored in other moments except rising edge of ENABLE. So, if the user changes one of these input parameters while ENABLE is already TRUE, these changes will not be taken in account. After correcting an error in one of these inputs, indicated by output STATUS, a new rising edge in input ENABLE is necessary to restart normal operation of FB DATALOGGER.**

3.1.2. FILE_NAME - Prefix for Naming Folders and CSV Files

- Type: STRING
- String size: from 1 to 20 characters
- Default value: empty

This input is a string that defines the prefix used for naming folders and CSV files. For details, see sections Folders and CSV Files in chapter Datalogger Working Principles.

The prefix is limited to 20 characters. If the prefix has more than 20 characters, only the first 20 characters are considered.

The following characters must not be used in this prefix:

- . (dot)
- / (slash)
- ~ (tilde)

> **ATTENTION:**
> **Different instances of FB DATALOGGER must have different values of FILE_NAME.**

3.1.3. STORAGE - Type of Non-Volatile Storage

- Type: DATALOGGER_STORAGE
- Possible values:
  - INTERNAL_MEMORY
  - MEMORY_CARD
  - USB_MASS_STORAGE
- Default value: INTERNAL_MEMORY

This input defines the type of non-volatile storage where the CSV files are saved in PLC. For details, see section Non-Volatile Storage Media in chapter Datalogger Working Principles.

3.1.4. FILE_RECORDS - Maximum Number of Records in a CSV File in Continuous Mode

- Type: UINT
- Range: 100 to 65535
- Default value: 0

This input defines how many records can be stored in a CSV file in continuous mode. This input is not considered in trigger mode.

For details, see section Continuous Mode in chapter Datalogger Working Principles.

3.1.5. NUMBER_OF_FILES - Maximum Number of CSV Files per Folder

- Type: UINT
- Range: 1 to 65535
- Default value: 0

This input defines how many CSV files can be saved inside a folder by an instance of FB DATALOGGER. For details, see sections Folders, CSV Files and Circular and Non-Circular Storage of Files, in chapter Datalogger Working Principles.

3.1.6. MODE - Datalogger Mode

- Type: DATALOGGER_MODE
- Possible values:
  - CONTINUOUS_MODE
  - TRIGGER_MODE
- Default value: CONTINUOUS_MODE

This input defines the operating mode of FB DATALOGGER. For details, see section Continuous and Trigger Modes in chapter Datalogger Working Principles.

3.1.7. CIRCULAR - Circular Storage of Files

- Type: BOOL
- Default value: FALSE

This input defines if storage of files is circular (TRUE) or non-circular (FALSE). For details, see section Circular and Non-Circular Storage of Files inside chapter Datalogger Working Principles.

3.1.8. TRIGGER_BEFORE - Number of Records before Trigger

- Type: UINT
- Range: 1 to 1000
- Default value: 0

This input defines how many records are saved in a CSV file before rising edge of TRIGGER input, in trigger mode. This input is not considered in continuous mode.

For details, see section Continuous and Trigger Modes in chapter Datalogger Working Principles.

3.1.9. TRIGGER_AFTER - Number of Records after Trigger

- Type: UINT
- Range: 1 to 1000
- Default value: 0

This input defines how many records are saved in a CSV file after rising edge of TRIGGER input, in trigger mode. This input is not considered in continuous mode.

For details, see section Continuous and Trigger Modes in chapter Datalogger Working Principles.

3.1.10. TRIGGER - Trigger Event

- Type: BOOL
- Default value: FALSE

The rising edge of this input defines a triggering event in trigger mode. For details, see section Continuous and Trigger Modes in chapter Datalogger Working Principles.

3.1.11. REC_INTERVAL - Time Interval between Records of CSV Files

- Type: TIME
- Range:
  - Bigger or equal than T#10ms in trigger mode
  - Bigger or equal than T#100ms in continuous mode
- Default value: T#0ms

This input defines the period used for sampling input values (INPUT001 ... INPUT256) and saving them in CSV files. For details, see section Behavior of REC_INTERVAL and Sample Time of FB DATALOGGER in chapter Datalogger Working Principles.

3.1.12. COMMA_DELIMITER - Types of CSV Delimiters

- Type: BOOL
- Default value: FALSE

This input must be TRUE to define CSV delimiters used in countries like US, and must be FALSE to define CSV delimiters used in countries like Brazil.

With COMMA_DELIMITER = TRUE:

- Comma is used as delimiter between columns of CSV files;
- Dot is used as a decimal point in value columns, and between seconds and milliseconds in TIME columns.

With COMMA_DELIMITER = FALSE:

- Semicolon is used as delimiter between columns of CSV files;
- Comma is used as a decimal point in value columns, and between seconds and milliseconds in TIME columns.

For details, see section Types of CSV Delimiters in chapter Datalogger Working Principles.

3.1.13. FILE_HEADER - Definition of Final Part of File Header

- Type: REFERENCE STRING
- String size: from 1 to 3327 characters
- Default value: empty

> **ATTENTION:**
> **Although the type of this input is "REFERENCE TO STRING", the user must pass a parameter of type STRING.**

This input defines a string with titles for the value columns of CSV files. It is also used to verify how many inputs values (see input parameters INPUT001 ... INPUT256) are considered by the FB.

The delimiters between titles for value columns must be in accordance with input parameter COMMA_DELIMITER (commas or semicolons). If this rule is not obeyed, an error message will be produced.

For details about this, see section Selection of Values to be Logged.

This string may contain up to 3327 characters. If it contains more than 3327 characters, an error message will be produced.

The maximum number of delimiters (commas or semicolons) inside the string must be 255, because maximum number of inputs is 256.

The average size of each title for a value column is 12 characters, considering maximum of 256 inputs (256 * 12 characters for titles + 255 characters for delimiters = 3327 characters).

3.1.14. INPUT001 ... INPUT256 - Value Inputs for Logging

- Type: REAL
- Default value: 0

These 256 inputs are the values logged by FB DATALOGGER.

For details, see sections Selection of Values to be Logged and Types of Values to be Logged in chapter Datalogger Working Principles.

3.2. Output Parameters

3.2.1. DONE - Completed Record Writing

- Type: BOOL

This output is mainly intended to inform that FB DATALOGGER has finished operations for writing records in CSV files. In some circumstances it is also used for indicating that FB DATALOGGER has found some kind of error (see output ERROR).

For more details, see section Timing Diagrams of Boolean Outputs (DONE, BUSY, ERROR) inside chapter Datalogger Working Principles.

3.2.2. BUSY - Busy Writing Records

- Type: BOOL

This output is mainly intended to inform that FB DATALOGGER is busy writing records to CSV files. In some circumstances it also is used for indicating that FB DATALOGGER is creating the directory and queue for writing files.

For more details, see section Timing Diagrams of Boolean Outputs (DONE, BUSY, ERROR) inside chapter Datalogger Working Principles.

3.2.3. ERROR - Error or Status Condition Active

- Type: BOOL

This output is TRUE when at least one error condition was found, that is, when output STATUS is different from NO_ERROR.

For more details, see section Timing Diagrams of Boolean Outputs (DONE, BUSY, ERROR) inside chapter Datalogger Working Principles.

3.2.4. STATUS - Error or Status Description

- Type: DATALOGGER_STATUS

This output describes an error conditions. The following values can be assumed:

- NO_ERROR: datalogger is working normally, without error conditions;
- INVALID_FILE_NAME: the FILE_NAME input is an empty string, or has invalid characters;
- INVALID_FILE_HEADER: the FILE_HEADER input has one of the following problems:
  - Is an empty string;
  - Has more than 3327 characters;
  - Defines less than 1 input;
  - Defines more than 256 inputs;
  - Defines one or more empty titles for inputs;
  - Contain at least one semicolon character when input COMMA_DELIMITER is TRUE;
  - Contain at least one comma character when input COMMA_DELIMITER is FALSE.
- INVALID_STORAGE: input STORAGE has an invalid value;
- INVALID_REC_INTERVAL: input REC_INTERVAL is out of allowed range;
- INVALID_MODE: input MODE has an invalid value;
- INVALID_TRIGGER_BEFORE: input TRIGGER_BEFORE is out of allowed range while input MODE is TRIGGER_MODE;
- INVALID_TRIGGER_AFTER: input TRIGGER_AFTER is out of allowed range while input MODE is TRIGGER_MODE;
- INVALID_FILE_RECORDS: input FILE_RECORDS is out of allowed range while input MODE is CONTINUOUS_MODE;
- INVALID_NUMBER_OF_FILES: input NUMBER_OF_FILES is out of allowed range;
- ERROR_NUMBER_OF_FILES_REACHED: this error happens when input CIRCULAR is FALSE and the number of CSV files created by the FB reached the limit given by input NUMBER_OF_FILES;
- ERROR_FILE_OPEN: error when FB was trying to open a file;
- ERROR_FILE_WRITE: error when FB was trying to write to a file;
- ERROR_FILE_CLOSE: error when FB was trying to close a file;
- ERROR_OVERLOAD: buffer overflow problem;
- ERROR_STORAGE_DEVICE_UNMOUNTED: the storage device specified by input STORAGE could not be mounted;
- ERROR_NOSPACE: there is no more space for CSV files in the storage device specified by input STORAGE;
- ERROR_RTC_READING: error when FB was trying to read date and time from real time clock;
- INTERNAL_ERROR: nonspecific internal error.

3.2.5. CURRENT_RECORDS - Number of Records Saved

- Type: UDINT

Number of records saved by FB DATALOGGER since last rising edge of input ENABLE.

3.2.6. MISSED_TRIGGERS - Number of Missed Trigger Events

- Type: UDINT

Number of trigger events (rising edges of input TRIGGER) that were missed, that is, that do not created a corresponding CSV file. This may happen when the interval between two consecutive rising edges of TRIGGER is less than TRIGGER_AFTER * REC_INTERVAL.

4. Application Example

The following figure shows an example with one instance of FB DATALOGGER.

ATTENTION:
Several inputs of the FB instance are not connected in this example. The unconnected inputs assume their default values, informed in section Input Parameters of chapter Input and Output Parameters of Function Block DATALOGGER.

```
PROGRAM Datalogger_Test_01
VAR
    DLOG_01 : DATALOGGER;
    ENABLE_01 : BOOL;
    TRIGGER_01 : BOOL;
    FILE_HEADER_01 : STRING := 'TEMPERATURE;PRESSURE';
    TEMPERATURE : REAL;
    PRESSURE : REAL;
    DONE_01 : BOOL;
    BUSY_01 : BOOL;
    ERROR_01 : BOOL;
    STATUS_01 : DATALOGGER_STATUS;
    CURRENT_RECORDS_01 : UDINT;
    MISSED_RECORDS_01 : UDINT;
END_VAR
```

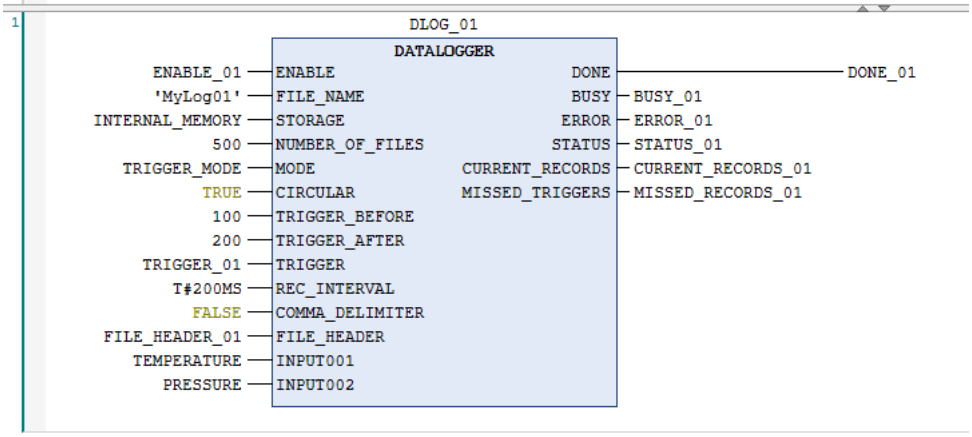


Figure 20: Example of application with FB DATALOGGER

The instance has the following features:

- Files are named MyLog01_nnnnn, where "nnnnn" is the sequential number;
- CSV files are stored in the internal FLASH memory;
- Each folder may contain 500 files;
- Trigger mode is used;
- Circular storage is used inside each folder;
- Each CSV file has 100 records before trigger and 200 records after trigger;
- Recording interval is 200 ms;
- Semicolons are used as CSV columns delimiter, and commas are used as decimal points;
- Two value inputs are recorded: TEMPERATURE and PRESSURE.

The following figure shows an example of CSV file captured using this application. Note that the file has 301 records after the header (TRIGGER_BEFORE + 1 + TRIGGER_AFTER). In this example, values of TEMPERATURE and PRESSURE were fixed with value 0.

	A	B	C	D	E
1	RECORD	DATE	TIME	TEMPERATURE	PRESSURE
2	-100	19/06/2019	16:19:09,674	0	0
3	-99	19/06/2019	16:19:09,874	0	0
4	-98	19/06/2019	16:19:10,074	0	0
5	-97	19/06/2019	16:19:10,274	0	0
6	-96	19/06/2019	16:19:10,474	0	0
7	-95	19/06/2019	16:19:10,674	0	0
8	-94	19/06/2019	16:19:10,874	0	0
9	-93	19/06/2019	16:19:11,074	0	0
10	-92	19/06/2019	16:19:11,274	0	0
11	-91	19/06/2019	16:19:11,473	0	0
95	-7	19/06/2019	16:19:28,274	0	0
96	-6	19/06/2019	16:19:28,473	0	0
97	-5	19/06/2019	16:19:28,673	0	0
98	-4	19/06/2019	16:19:28,874	0	0
99	-3	19/06/2019	16:19:29,073	0	0
100	-2	19/06/2019	16:19:29,274	0	0
101	-1	19/06/2019	16:19:29,474	0	0
102	0	19/06/2019	16:19:29,573	0	0
103	1	19/06/2019	16:19:29,673	0	0
104	2	19/06/2019	16:19:29,874	0	0
105	3	19/06/2019	16:19:30,074	0	0
106	4	19/06/2019	16:19:30,273	0	0
107	5	19/06/2019	16:19:30,474	0	0
291	189	19/06/2019	16:20:07,274	0	0
292	190	19/06/2019	16:20:07,474	0	0
293	191	19/06/2019	16:20:07,674	0	0
294	192	19/06/2019	16:20:07,873	0	0
295	193	19/06/2019	16:20:08,074	0	0
296	194	19/06/2019	16:20:08,274	0	0
297	195	19/06/2019	16:20:08,474	0	0
298	196	19/06/2019	16:20:08,674	0	0
299	197	19/06/2019	16:20:08,874	0	0
300	198	19/06/2019	16:20:09,074	0	0
301	199	19/06/2019	16:20:09,273	0	0
302	200	19/06/2019	16:20:09,474	0	0

Figure 21: Example of file MyLog01_00001.csv for application with FB DATALOGGER