

## Operating Guide

# How to build software with the data log

## Summary

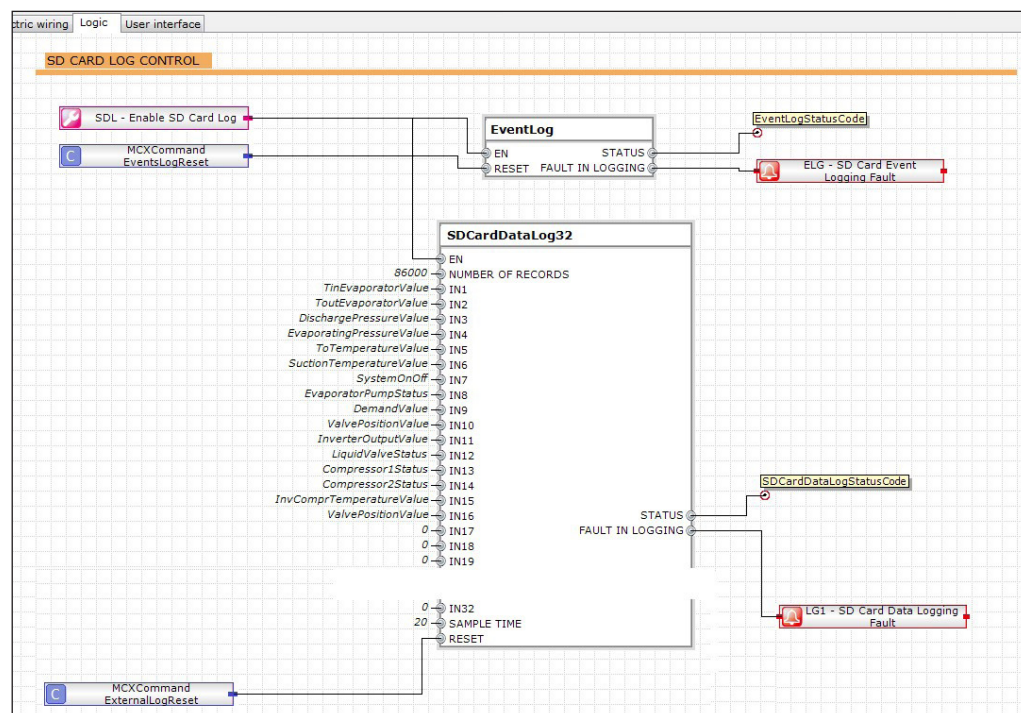
In software made using MCXDesign, it is possible to add a data log function. This function only works with MCX061V and MCX152V.

The data is saved in internal memory or/and in SD card memory and can be read through a WEB connection or by a PC using a decode program.

## Description

### MCXDesign part

1. In the “**LogLibrary**” there are three bricks that enable the addition of data logging to software made using MCXDesign: one brick is for events and the others enable the selection of variables and of memory for storing the data.
2. Software with data logging looks like the image below:



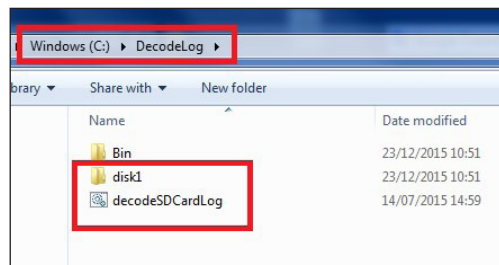
**Note:** The data logging feature is only available in the MCX hardware (it cannot be simulated using the software simulation).

3. The “**EventLog**” brick and “**SDCardDataLog32**” brick save the file to the SD memory, and the “**MemoryDataLog16**” brick saves the file to MCX internal memory.

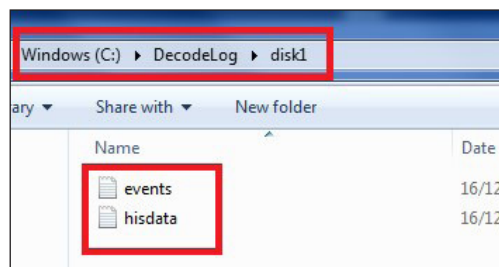
**Note:** For additional information, please refer to the bricks’ help.

### Reading the file through the decode program

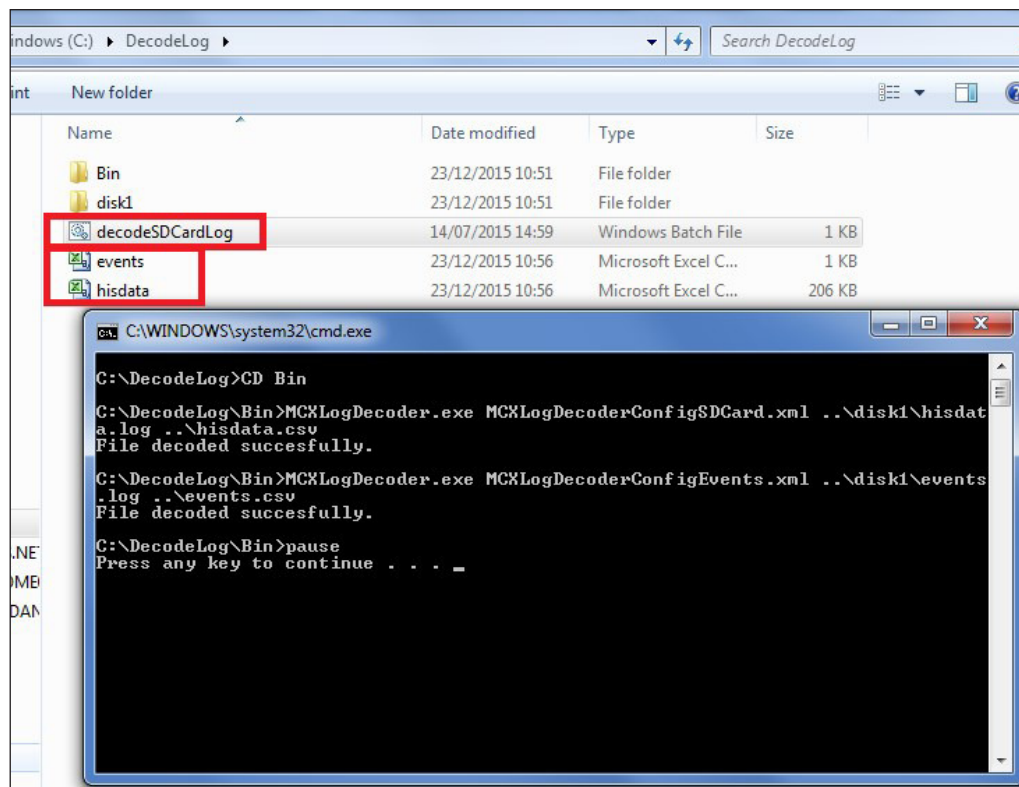
1. The files saved on the SD card can be read through a WEB connection or using a batch file. However, the file saved on the internal memory can be read only through WEB.
2. To read the files on the SD card using a decode program, download the **"DecodeLog"** folder available on the MCX site and save it to the C disk:



3. Extract the memory card from the MCX and copy and paste the files to the SD card in the **"DecodeLog/Disk1"** folder:



4. From the **"DecodeLog"** folder, run the batch file **"decodeSDCardLog"**. This will generate the .csv files with encoded data:



5. Events are recorded in the events.csv file. There are six columns:
  - a. Event time: the time of the event (start alm, stop alm, parameters change and RTC change)
  - b. EventNodeID: the ID of the MCX
  - c. EventType: a numerical description of the event type
    - -2: Reset of MCX history alarm
    - -3: RTC set
    - -4: Start alarm
    - -5: Stop alarm
    - 1000: Parameters change (note: the change can be detected only when it is made through a user interface)
  - d. Var1: a numerical description of the variable. To decrypt it, open the "AGFDefine.c" file in the "App" folder of the MCXDesign software. In this file there are two sections with an ID indication: one is for the parameters and the other is for the alarm. If the event type is 1000, refer to the index parameters' list; if the event type is -4 or -5, refer to the index alarms' list. These lists contain the variable names corresponding to each ID (not to the variable description – for the variable description, refer to MCXShape).

The screenshot displays two windows. The left window is 'events - Microsoft Excel', showing a table of event data. The right window is 'AGFDefine.c', showing a list of variables and their IDs.

EventTime	EventNodeID	EventType	Var1	Var2
15/12/2015 16:50	1	-5	12	0
15/12/2015 16:50	1	-5	13	0
15/12/2015 16:50	1	1000	12	13762780
15/12/2015 16:51	1	-4	6	0
15/12/2015 16:51	1	-4	7	0
15/12/2015 16:51	1	-5	7	0
15/12/2015 16:51	1	-5	6	0
15/12/2015 17:17	1	-4	3	0
15/12/2015 17:18	1	-5	3	0
15/12/2015 17:18	1	1000	12	14418130
16/12/2015 08:31	1	-4	6	0
16/12/2015 08:31	1	-4	7	0
16/12/2015 08:31	1	-5	7	0
16/12/2015 08:31	1	-5	6	0
16/12/2015 09:20	1	1000	12	13762780
16/12/2015 09:47	1	-2	0	0
16/12/2015 09:47	1	-2	0	0
16/12/2015 09:49	1	-4	6	0
16/12/2015 09:49	1	-4	7	0
16/12/2015 09:49	1	-5	6	0
16/12/2015 09:49	1	-5	7	0
16/12/2015 15:21	1	-4	6	0
16/12/2015 15:21	1	-4	7	0
16/12/2015 15:21	1	-5	7	0
16/12/2015 15:21	1	-5	6	0
16/12/2015 15:21	1	1000	12	14418130

The 'AGFDefine.c' file shows a list of variables and their IDs. The 'Index alarms list' section is highlighted in green.

```

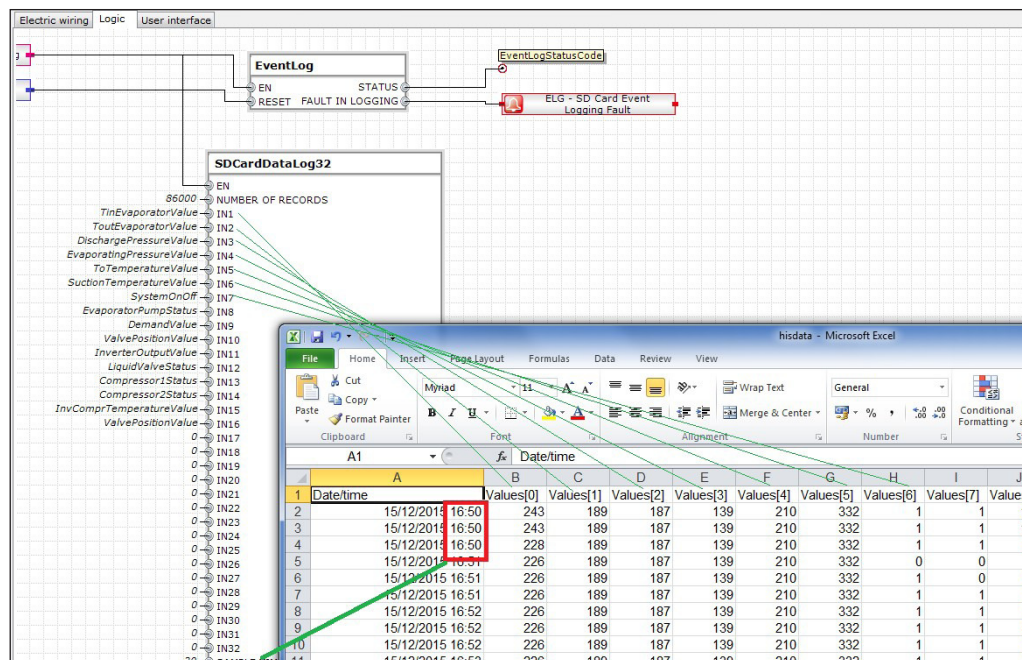
// Index alarms list
#define AL_EV_In_ID 0
#define AL_T_Ev_Out_ID 1
#define AL_T_Suction_Cl_ID 2
#define AL_T_InverterCompr_ID 3
#define AL_F_Discharge_ID 4
#define AL_F_Evaporating_ID 5
#define AL_Gen_ID 6
#define AL_External_ID 7
#define AL_EXV_1_Configuration_ID 8
#define AL_EXV_1_BatteryFault_ID 9
#define AL_EXV_2_Configuration_ID 10
#define AL_EXV_2_BatteryFault_ID 11
#define AL_EM_DataLogging_Fault_ID 12
#define AL_EM_EventLogging_Fault_ID 13
  
```

The screenshot shows the MCXShape software interface. The 'Main Menu' is on the left, and the 'Parameters' tab is selected on the right. The 'Parameters' tab displays a table of variables and their descriptions.

Label	Description	Variable Name
CUM	Serial settings (Modbus)	Params_Sys_SerialSet
<b>GENERAL &gt; PASSWORD</b>		
L01	Password level 1	Params_Password1
L02	Password level 2	Params_Password2
L03	Password level 3	Params_Password3
<b>PARAMETERS &gt; UNIT CONFIG</b>		
C01	Compressors for Circuit	Params_CompressorsForCircuit
<b>PARAMETERS &gt; TEMPERATURE CO...</b>		
SE	Setpoint	Params_Set
dIF	Differential	Params_Diff
<b>PARAMETERS &gt; COMPRESSOR TIM...</b>		
CT1	Automatic Rotation	Params_CompE_Rotation

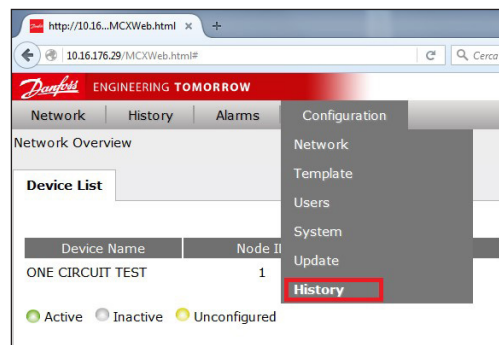
- f. Var2: used to record the parameter value before and after the change. This number is a double integer; in the high part there is the new parameter value and in the low part there is the old value.
- g. Var3: not used.

- Recorded in the hisdata.csv file are all the variables defined in MCXDesign in relation to the sample time in the order defined in the brick:



## Reading the file in WEB

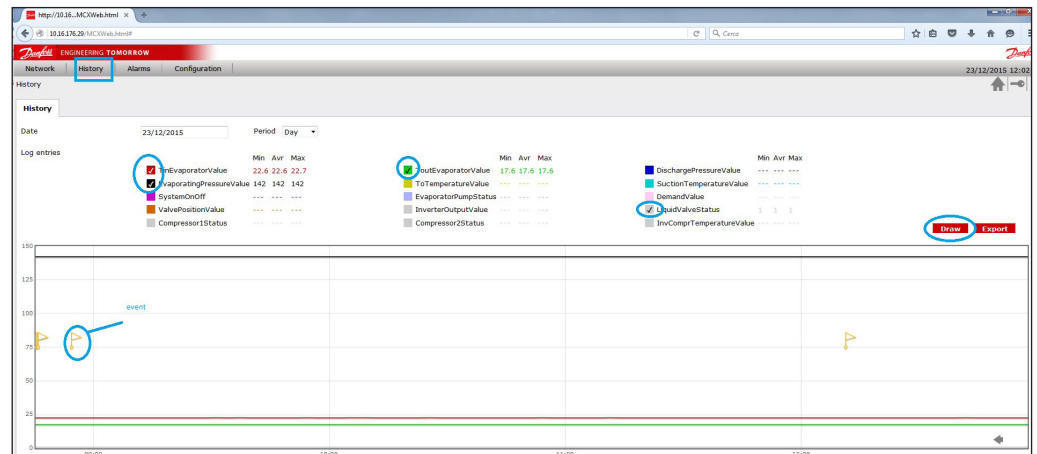
- To read these files in WEB, use the latest MCXWeb pages available at the MCX web site. In the Configuration/History menu, set up the variables to monitor (max. 15).



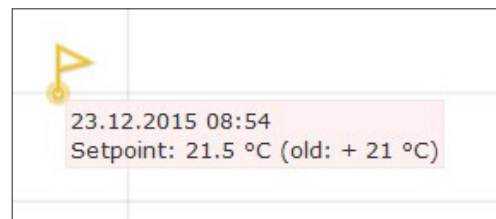
- In the Configuration/History menu you must define:
  - Node: not important.
  - Parameters: can be chosen only from the variables stored in the log file. This setting is used to take information about the variable's decimal point and unit of measure.
  - Colour: defines the line colour in the graph.
  - File: defines the file where the variable value is taken from.
  - Position: the position (column) of the variable in the file (see also point 9):

Index	Node	Parameter	Color	File	Position
1	1 - ONE CIRCUIT TEST	TinEvaporatorValue	#CC0000	1:/hisdata.log	1
2	1 - ONE CIRCUIT TEST	ToutEvaporatorValue	#008000	1:/hisdata.log	2
3	1 - ONE CIRCUIT TEST	DischargePressureValue	#0000CC	1:/hisdata.log	3
4	1 - ONE CIRCUIT TEST	EvaporatingPressureValue	#000000	1:/hisdata.log	4
5	1 - ONE CIRCUIT TEST	ToTemperatureValue	#FF0000	1:/hisdata.log	5

3. From the history menu, data can be graphed and exported in a .csv file:
  - a. Choose the variable to graph.
  - b. Define "Data" and "Period".
  - c. Draw.
  - d. Export to create a .csv file.



**Note:** The graph also has events (yellow flags); use the mouse to click a flag in order to get additional information about the event.



## Danfoss A/S

Climate Solutions • danfoss.com • +45 7488 2222

Any information, including, but not limited to information on selection of product, its application or use, product design, weight, dimensions, capacity or any other technical data in product manuals, catalogues descriptions, advertisements, etc. and whether made available in writing, orally, electronically, online or via download, shall be considered informative, and is only binding if and to the extent, explicit reference is made in a quotation or order confirmation. Danfoss cannot accept any responsibility for possible errors in catalogues, brochures, videos and other material. Danfoss reserves the right to alter its products without notice. This also applies to products ordered but not delivered provided that such alterations can be made without changes to form, fit or function of the product.

All trademarks in this material are property of Danfoss A/S or Danfoss group companies. Danfoss and the Danfoss logo are trademarks of Danfoss A/S. All rights reserved.