VDE Test Report

Report No. ....:	294956-TL2-1
VDE File No. ....:	5016540-4970-0006/294956
Date of issue..:	2022-10-21
Laboratory...:	**VDE** Testing and Certification Institute
Address ....:	Merianstrasse 28 63069 Offenbach/Main; Germany
Testing location/ address...........................:	**VDE** Testing and Certification Institute Merianstrasse 28 63069 Offenbach/Main; Germany
Applicant's name.....................................:	NXP Semiconductors Czech Republic s.r.o.
Applicant's address ....:	1. Maje 1009; 756 61 Roznov pod Radhostem; Czech Republic
Applied standard(s) ....:	DIN EN 60730-1 (VDE 0631-1):2021-06; DIN EN 60335-1 (VDE 0700-1):2020-08; EN 60730-1:2016+A1:2019 EN 60335-1:2012+AC+A11+A13+A1+A2+A14:2019 EN 60335-1:2012/A15:2021 IEC 60730-1:2013 IEC 60730-1:2013/AMD1:2015 IEC 60730-1:2013/AMD2:2020 IEC 60335-1:2010 IEC 60335-1:2010/AMD1:2013 IEC 60335-1:2010/AMD2:2016
Test item description ....:	Micro controller DSC series
Trade Mark ....:	NXP

Type reference(s)....................................:	File Name	Revision
	iec60730b_dsc_reg.c	4.1
	iec60730b_dsc_pc.c	4.1
	iec60730b_dsc_pc_object1.c	4.1
	iec60730b_dsc_pc_object2.c	4.1
	iec60730b_dsc_clock.c	4.1
	iec60730b_dsc_flash.c	4.1
	iec60730b_dsc_ram.c	4.1
	iec60730b_dsc_dio.c	4.1
	iec60730b_dsc_dio_ext.c	4.1
	iec60730b_dsc_aio.c	4.1
	IEC60730B_DSC_EX_CW_LDM_v4_3.lib*	4.3

	IEC60730B_DSC_EX_CW_SDM_v4_3.lib*	4.3
	IEC60730B_DSC_EF_CW_LDM_v4_3.lib*	4.3
	IEC60730B_DSC_EF_CW_SDM_v4_3.lib*	4.3
Remark:	The file marked with * is object code file.	

Test sample condition ....:	☒	Non-damaged sample	
	Remark:	*/*	
Sample entry date....................................:	2022-10-21		
Date (s) of performance of tests ....:	2022-10-21		

Tested by...:	
Name, Signature ....:	J. Schildbach (Authorization of test report)
Function..:	Testing engineer
Verified by...:	
Name, Signature ....:	K. Tas
Function..:	Technical Certification Officer

Factory(ies)...:	NXP Semiconductors Czech; Republic s.r.o. 1. Maje 1009; 756 61 ROZNOV POD RADHOSTEM CZECH REPUBLIK

Possible test case verdicts:	
Test case does not apply to the test object :	N/A
Test object does meet the requirement......:	P (Pass)
Test object does not meet the requirement :	F (Fail)

Final Verdict:	☒	**P**	☐	**F**
Remark...:	*/*			

Environmental conditions (if applicable)	Ambient temperature	Atmospheric pressure	Relative humidity
Rated values...:	15-35 °C	860-1060 hPa	30-60 %
Verified values ....:	N/A	Range confirmed by: Deutscher Wetterdienst (Meteorological service)	N/A

General Remarks:
Conformity statement: The VDE decision rule for the statement of conformity is in accordance with IEC Guide 115:2021 procedure 2

General Remarks:
All testing was done with revision 4.1, with source code files and object code files with identical test results.   All self-diagnostic routines are executable after reset and during runtime. For variable memory self-diagnostics, the user can select between MARCH-X and MARCH-C algorithm.

Naming conventions for object code files:

- **Standard – IEC60730**
- Library class compliance according to the IEC60730 standard – **B**
- The MCU core, for which the library was compiled – **DSC**
- The MCU extension type, for which the library was compiled – **EX**, or **EF**
- The type of compiler/IDE – **CW** (Code Warrior)
- The type of data model, for which the library was compiled – **LDM**, or **SDM**
- Identification of the library version – **v4.3**
- File extension – ***.lib**

**IEC60730B_DSC_EX_CW_LDM_v4_3.lib**

The object code files revision 4.3 can be verified by its SHA-256 Hash-Tag shown by an example on next page as follows:

File Name:	SHA-256 Hash-Tag:
IEC60730B_DSC_EX_CW_LDM_v4_3.lib*	6654DEA152AC38E8B05D1918B6912C836F7C83FE8DD74A9D4C29AC13388973C4
IEC60730B_DSC_EX_CW_SDM_v4_3.lib*	705D50150A7AA12BBFCF8C3849925C7D92FF5D934357C7F5900004342FFD64BE
IEC60730B_DSC_EF_CW_LDM_v4_3.lib*	FE1B427C565B40FF7A465546FE1C6995DF76AFF92A99A39FFD927FCC72B18FBA
IEC60730B_DSC_EF_CW_SDM_v4_3.lib*	58EB2EDBA7EFB8DB19662D3A60078D00F0C633B77349783AD58B29D13391E4F6

1. Download from internet a utility like "MDS_and_SHA_Checksum_Utility.exe" via the following link:
https://download.cnet.com/MD5-SHA-Checksum-Utility/3000-2092_4-10911445.html.
(Another example is the link https://www.nirsoft.net/utils/hash_my_files.html. Click to "Download HashMyFiles".)

2. Start the utility.
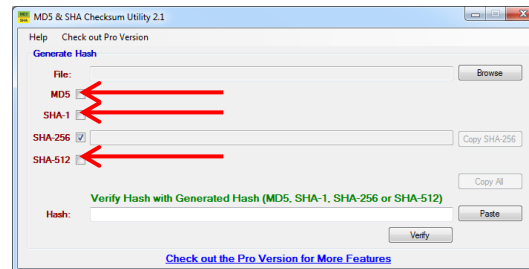
Delete the hooks for "MDS", "SHA-1" and SHA-512 (see red arrows).
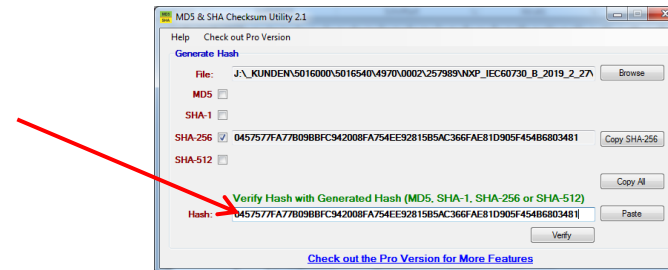
VDE Testing and Certification Institute

3. In the windows file explorer mark the downloaded object code file and move it the line "File" (see red arrow).
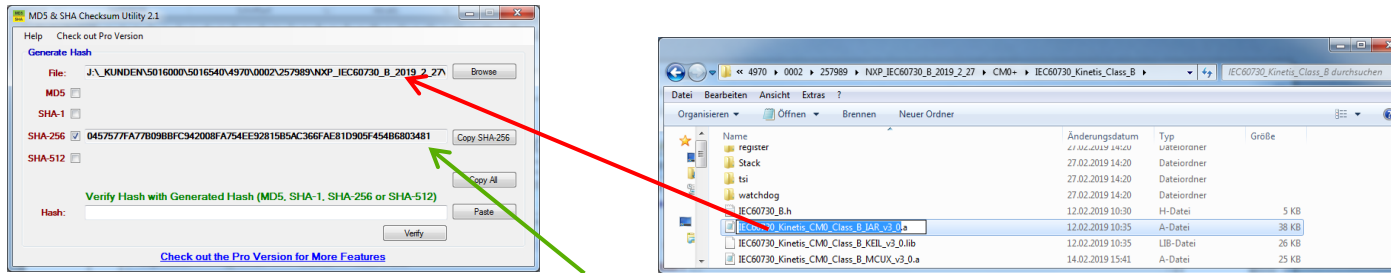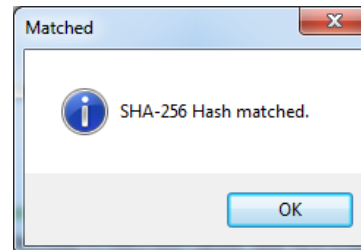

The "SHA-256" Hash-Tag for this file will be generated directly (see green arrow).

4. Copy the corresponding Hash-Tag from this test report and insert it to the line "Hash" (see red arrow).

5. Now press the button "Verify". If both Hash-Tags are identically the window below will pop up.

Performed Tests

IEC 60335-1 (Annex R)

Clause	Requirement + Test	Result – Remark	Verdict
R	ANNEX R (NORMATIVE) (60335-1) SOFTWARE EVALUATION		—
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2 validated in accordance with the requirements of this annex	Self-test routines for software of class R.1	P
R.1	Programmable electronic circuits using software		—
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2 constructed so that the software does not impair compliance with the requirements of this standard		P
R.2	Requirements for the architecture		—
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2 use measures to control and avoid software-related faults/errors in safety-related data and safety-related segments of the software		P
R.2.1.1	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.2 have one of the following structures:		—
	- single channel with periodic self-test and monitoring		N/A
	- dual channel (homogenous) with comparison		N/A
	- dual channel (diverse) with comparison		N/A
	Programmable electronic circuits requiring software incorporating measures to control the fault/error conditions specified in table R.1 have one of the following structures:		—
	- single channel with functional test		P
	- single channel with periodic self-test		P
	- dual channel without comparison		N/A
R.2.2	Measures to control faults/errors		—
R.2.2.1	When redundant memory with comparison		N/A

	is provided on two areas of the same component, the data in one area is stored in a different format from that in the other area		
R.2.2.2	Programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.2 and that use dual channel structures with comparison, have additional fault/error detection means for any fault/errors not detected by the comparison		N/A
R.2.2.3	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, means are provided for the recognition and control of errors in transmissions to external safety-related data paths		N/A
R.2.2.4	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, the programmable electronic circuits incorporate measures to address the fault/errors in safety-related segments and data indicated in table R.1 and R.2 as appropriate		P
R.2.2.5	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, detection of a fault/error occur before compliance with clause 19 is impaired	Self-test routines only; compliance to clause 19 has to be insured by the user of the self-test routines	N/A
R.2.2.6	The software is referenced to relevant parts of the operating sequence and the associated hardware functions		P
R.2.2.7	Labels used for memory locations are unique		P
R.2.2.8	The software is protected from user alteration of safety-related segments and data		P
R.2.2.9	Software and safety-related hardware under its control is initialized and terminates before compliance with clause 19 is impaired	Self-test routines only; compliance to clause 19 has to be insured by the user of the self-test routines	N/A
R.3	Measures to avoid errors		—
R.3.1	General		—

	For programmable electronic circuits with functions requiring software incorporating measures to control the fault/error conditions specified in table R.1 or R.2, the following measures to avoid systematic fault in the software are applied		—
	Software that incorporates measures used to control the fault/error conditions specified in table R.2 is inherently acceptable for software required to control the fault/error conditions specified in table R.1	Class R.1 only	N/A
R.3.2	Specification		—
R.3.2.1	Software safety requirements:	Software Id: 4.1	P
	The specification of the software safety requirements includes the descriptions listed		P
R.3.2.2	Software architecture		—
R.3.2.2.1	The specification of the software architecture includes the aspects listed - techniques and measures to control software faults/errors (refer to R.2.2); - interactions between hardware and software; - partitioning into modules and their allocation to the specified safety functions; - hierarchy and call structure of the modules (control flow); - interrupt handling; - data flow and restrictions on data access; - architecture and storage of data; - time-based dependencies of sequences and data	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0	P
R.3.2.2.2	The architecture specification is validated against the specification of the software safety requirements by static analysis		P
R.3.2.3	Module design and coding		—
R.3.2.3.1	Based on the architecture design, software is suitably refined into modules		P
	Software module design and coding is implemented in a way that is traceable to the software architecture and requirements		P
R.3.2.3.2	Software code is structured		P
R.3.2.3.3	Coded software is validated against the module specification by static analysis		P
	The module specification is validated against the architecture specification by static analysis	Reviews and source code walk through	P
R.3.3.3	Software validation		—
	The software is validated with reference to the requirements of the software safety requirements specification		P

	Compliance is checked by simulation of:		—
	- input signals present during normal operation		P
	- anticipated occurrences		P
	- undesired conditions requiring system action		P

IEC 60730-1 (Software)

Clause	Requirement + Test	Result - Remark	Verdict
H.6	Classification, additions		—
H.6.18	Class of control function (A, B,C)........................		—
H.7	Information in addition to Table 1 provided:		—
	66 - Software sequence documentation; clause: H.11.12.2.9; method: X	Self-diagnostiv modules acc. Table R.1/H.1 only	N/A
	67 - Program documentation; clause: H.11.12.2.9, H.11.12.2.12; method: X		P
	68 - Software fault analysis; clause: H.11.12, H.27.1.1.4; method: X		P
	69 - Software class(es) and structure; clause: H.11.12.2, H.11.12.3, H.27.1.2.2.1, H.27.1.2.3.1; method: D		P
	70 - Analytical measures and fault/error control techniques employed; clause: H.11.12.1.2, H.11.12.2.2, H.11.12.2.4; method: X....................		P
	71 - Software fault/error detection time(s) for controls with software Classes B or C; clause: H.2.17.10, H.11.12.2.6; method: X...........		P
	72 - Control response(s) in case of detected fault/error; clause: H.11.12.2.7; method: X...........		P
	93 – Maximum number of reset actions within a time period; clause H.11.12.4.3.6, H.11.12.4.3.4; method: D		N/A
	94 – Number of remote reset actions; clause H.17.1.4.3; method: X		N/A
	m – Controls with software classes B or C had information provided for safety-related segments of the software. Information on the non-safety related segments was sufficient to establish that they did not influence safety-related segments.....		N/A
	n – Software sequence was documented and, together with the operating sequence, included a description of the control system philosophy, the control flow, data flow and the timings.................		N/A

	o - Safety-related data and safety-related segments of the software sequence, the malfunction of which could result in non-compliance with the requirements of Clauses 17, 25, 26 and 27, are identified		P
	– Included the operating sequence		N/A
	– Software fault analysis was related to the hardware fault analysis in Clause H.27		N/A
	q - Programming documentation was supplied in a programming design language declared by the manufacturer		P
	r – Different software classes applied to different control functions		N/A
	s - Measures declared are chosen by manufacturer from the requirements of Clauses H.11.12.1.2 to H.11.12.2.4 inclusive		N/A
H.11	Constructional requirements		—
H.11.12	Controls using software		—
	Controls using software were so constructed that the software did not impair control compliance with the requirements of this standard		P
H.11.12.1	Requirements for the architecture		—
H.11.12.1.1	Control functions with software class B or C use measures to control and avoid software-related faults/errors in safety-related data and safety-related segments of the software, as detailed in H.11.12.1.2 to H.11.12.3 inclusive		P
H.11.12.1.2	Control functions with software class C have one of the following structures:		—
	– single channel with periodic self-test and monitoring (H.2.16.7)		N/A
	– dual channel (homogenous) with comparison (H.2.16.3)		N/A
	– dual channel (diverse) with comparison (H.2.16.2)		N/A
	Control functions with software class B have one of the following structures:		—
	– single channel with functional test (H.2.16.5)		P
	– single channel with periodic self-test (H.2.16.6)		P

	– dual channel without comparison (H.2.16.1)		N/A
H.11.12.1.3	Other structure permitted with equivalent level of safety to those in H.11.12.1.2.............................		N/A
H.11.12.2	Measures to control faults/errors		—
H.11.12.2.1	Redundant memory with comparison provided on two areas of the same component: data stored in different formats		N/A
H.11.12.2.2	Software class C using dual channel structures with comparison: additional fault/error detection means		N/A
H.11.12.2.3	Software class B or C: means for recognition and control of errors in transmission to external safety-related data paths: Means took into account errors of data, addressing, transmission timing and sequence of protocol		N/A
H.11.12.2.4	Software class B or C: within the control, measures are taken to address the fault/errors in safety-related segments and data indicated in Table H.1 and identified in Table 1 requirement 68.		P
H.11.12.2.5	Measures others than those specified in H.11.12.2.4 utilized to satisfy the requirements listed in Table H.1		N/A
H.11.12.2.6	Software fault/error detection:		—
	– occur not later than declared time(s), Table 1, requirement 71		P
	– acceptability of declared time(s): evaluated during fault analysis of the control		P
H.11.12.2.7	For controls with functions, classified as Class B or C, detection of fault/error:		—
	– results in the response declared in Table 1, requirement 72		P
	– for Class C: independent means capable of performing this response provided		N/A
H.11.12.2.8	Class C, dual channel structure, loss of dual channel capability: deemed to be an error		N/A
H.11.12.2.9	Software referenced:		—

Report No.:	294956-TL2-1	Page	12	of	25

	– to relevant parts of the operating sequence		P
	– to the associated hardware functions		P
H.11.12.2.10	Labels used for memory locations are unique		P
H.11.12.2.11	Software protected from user alteration of safety-related segments and data		P
H.11.12.2.12	Software and safety-related hardware under its control is initialized to and terminates at a declared state, Table 1, requirement 66...............		P
H.11.12.3	Measures to avoid errors		—
H.11.12.3.1	For controls with software class B or C the measures shown in Figure H.1 to avoid systematic faults are applied		P
	Other methods utilized that incorporate disciplined and structured processes including design and test phases		N/A
H.11.12.3.2	Specification		—
H.11.12.3.2.1	Software safety requirements		—
H.11.12.3.2.1.1	The specification of the software safety requirements includes:		—
	• A description of each safety related function to be implemented, including its response time(s): - functions related to the application including their related software classes - functions related to the detection, annunciation and management of software or hardware faults		P
	• A description of interfaces between software and hardware		P
	• A description of interfaces between any safety and non-safety related functions		N/A
H.11.12.3.2.2	Software architecture		—
H.11.12.3.2.2.1	The description of software architecture include the following aspects:		—
	• Techniques and measures to control software faults/errors (refer to H.11.12.2)		P

	• Interactions between hardware and software		P
	• Partitioning into modules and their allocation to the specified safety functions		P
	• Hierarchy and call structure of the modules (control flow)		P
	• Interrupt handling		P
	• Data flow and restrictions on data access		P
	• Architecture and storage of data		P
	• Time based dependencies of sequences and data		P
H.11.12.3 .2.2.2	The architecture specification is verified against the specification of the software safety requirements by static analysis		P
H.11.12.3 .2.3	Module design and coding		—
H.11.12.3 .2.3.1	Software is suitably refined into modules. Software module design and coding are implemented in a way that is traceable to the software architecture and requirements. The module design specified:		P
	– function(s)		P
	– interfaces to other modules		N/A
	– data		P
H.11.12.3 .2.3.2	Software code is structured		P
H.11.12.3 .2.3.3	Coded software is verified against the module specification, and the module specification is verified against the architecture specification by static analysis		P
H.11.12.3 .2.4	Design and coding standards		P
	Program design and coding standards is used during software design and maintenance		P
	Coding standards :		—
	– specified programming practice		P
	– proscribed unsafe language features		P

	– specify procedures for source code documentation		P
	– specify data naming conventions		P
H.11.12.3.3	Testing		—
H.11.12.3.3.1	Module design (software system design, software module design and coding)		—
H.11.12.3.3.1.1	A test concept with suitable test cases is defined based on the module design specification.		P
H.11.12.3.3.1.2	Each software module is tested as specified within the test concept		P
H.11.12.3.3.1.3	Test cases, test data and test results are documented		P
H.11.12.3.3.1.4	Code verification of a software module by static means includes such techniques as software inspections, walk-throughs, static analysis and formal proof		P
	Code verification of a software module by dynamic means includes functional testing, white-box testing and statistical testing		P
H.11.12.3.3.2	Software integration testing		—
H.11.12.3.3.2.1	A test concept with suitable test cases is defined based on the architecture design specification		P
H.11.12.3.3.2.2	The software is tested as specified within the test concept		P
H.11.12.3.3.2.3	Test cases, test data and test results are documented		P
H.11.12.3.3.3	Software validation		—
H.11.12.3.3.3.1	A validation concept with suitable test cases is defined based on the software safety requirements specification		P
H.11.12.3.3.3.2	The software is validated with reference to the requirements of the software safety requirements specification as specified within the validation concept		P
	The software is exercised by simulation or stimulation of:		—

	• input signals present during normal operation		P
	• anticipated occurrences		P
	• undesired conditions requiring system action		P
H.11.12.3.3.3.4	Test cases, test data and test results are documented		P
H.11.12.3.4	Other Items		—
H.11.12.3.4.1	Equipment used for software design, verification and maintenance was qualified appropriately and demonstrated to be suitable for purpose in manifold applications		P
H.11.12.3.4.2	Management of software versions: All versions are uniquely identified for traceability		P
H.11.12.3.4.3	Software modification		—
H.11.12.3.4.3.1	Software modifications are based on a modification request which details the following:		N/A
	• the hazards which may be affected		N/A
	• the proposed change		N/A
	• the reasons for change		N/A
H.11.12.3.4.3.2	An analysis is carried out to determine the impact of the proposed modification on functional safety.		N/A
H.11.12.3.4.3.3	A detailed specification for the modification is generated including the necessary activities for verification and validation, such as a definition of suitable test cases		N/A
H.11.12.3.4.3.4	The modification is carried out as planned		N/A
H.11.12.3.4.3.5	The assessment of the modification is carried out based on the specified verification and validation activities.		N/A
H.11.12.3.4.3.6	All details of modification activities are documented		N/A
H.11.12.3.5	For class C control functions: One of the combinations (a–p) of analytical measures given in the columns of table H.9 is used during hardware development...		N/A

H.11.12.4	Remotely actuated control functions		—
H.11.12.4.1.1	Data Exchange – General – Remotely actuated control functions are connected to separate, independent devices, which may themselves contain control functions or provide other information and any data exchange between these devices does not compromise the integrity of class B control function or class C control function.		N/A
H.11.12.4.1.2	Type of data - Message types for data exchange in a control function or functions are allocated to class A control function, class B control function or class C control function. The safety or protective relevance or influence, message types or data exchange are allocated only to class B control function or class C control functions, see Table H.10.		N/A
H.11.12.4.1.3.1	Communication of Safety Related Data – Transmission – Safety relevant data is transmitted authentically concerning:		N/A
	– data corruption		N/A
	– address corruption		N/A
	– wrong timing or sequence		N/A
	Data variation or corrupted data did not lead to an unsafe state		N/A
	Before transmitted data was used it was ensured that data corruption, address corruption and wrong timing or sequence are addressed using the measures as given in Annex H.		N/A
	The following failure modes are addressed:		—
	– permanent "auto-sending" or repetition,		N/A
	– interruption of data transfer		N/A
H.11.12.4.1.3.2	Access to data exchange - All types of access to class B control function or class C control function related data exchange systems is clearly restricted		N/A
	Adequate hardware/software measures are taken to ensure no unauthorized access to the control functions (class B and C; operating data, configuration parameters and/or software modules)		N/A

H.11.12.4.1.3.3	For class B and class C software revisions the requirements of H.11.12.3 and hardware configuration management are applied and the control maintains its protective functions		N/A
H.11.12.4.1.4	Remotely actuated control function operation have the duration or limits set before switching on except when automatic switching off is realized at the end of a cycle or the system is designed for permanent operation.		N/A
H.11.12.4.2	Priority of remotely actuated control functions over control functions does not lead to a hazardous condition.		N/A
H.11.12.4.3.1	Remote reset action is manually initiated.		N/A
	Reset functionality initiated by a hand-held device required at least two manual actions to activate		N/A
H.11.12.4.3.2	Reset functions are capable of resetting the system as intended		N/A
H.11.12.4.3.3	Unintended resets from safe state do not occur.		N/A
H.11.12.4.3.4	Any fault of the reset function does not cause the control or controlled function to result in a hazardous condition, and was evaluated for its Class B classification		N/A
H.11.12.4.3.5	For reset functions initiated by manual action not in visible sight of the appliance, the following additional requirements apply:		N/A
	– the actual status and relevant information of the process under control is visible to the user before, during and after the reset action;		N/A
	– the maximum number of reset actions within a time period is declared. Following this, any further reset is denied unless the appliance is physically checked		N/A
H.11.12.4.3.6	The reset function is evaluated on the final application.		N/A
	Manual switching of a thermostat or device with similar function that activates a reset is declared by the manufacturer and is suitable in the final application...		N/A
H.27.1.2	Protection against internal faults to ensure functional safety		—

H.27.1.2.1	Design and construction requirements		—
H.27.1.2.1.1	Fault avoidance and fault tolerance		—
	Controls incorporating control functions of class B or C are designed according to H.27.1.2 taking into account the failure modes of Cl. H.11.12 for software		P
	Systematic errors are avoided		P
	Random faults are dealt with by a proper system configuration		P
	Functional analysis of the application resulted in a structured design with:		P
	– Control flow		P
	– Data flow		P
	– Time related functions required by the application		P
	For custom-chips special attention was made to minimize systematic errors		N/A
	System configuration was failsafe or:		N/A
	Incorporated components with direct safety-critical functions guarded by safeguards that cause a completely independent safety shut-down in accordance to H.11.12 software class B or C		N/A
	- safeguards are built into hardware and,		N/A
	- safeguards are supplemented by software		N/A
	Time slot monitoring is sensitive to both an upper and a lower limit of the time interval.		N/A
	Faults resulting in a shift of the upper and/or lower limit are taken into account.		N/A
	In a class C control function when a single fault in a primary safeguard can render the safeguard inoperative, a secondary safeguard is provided		N/A
	The reaction time of the secondary safeguard is in accordance with Clause H.27.1.2.3.		N/A
H.27.1.2.1.2	Documentation		—

	The documentation was based on H.11.12.3.2		P
	The functional analysis of the control and the safety related programs under its control are documented in a clear hierarchical way in accordance with the safety philosophy and the program requirements.		P
	Documentation provided for assessment included:		—
	• A description of the system philosophy, the control flow, data flow and timings.		P
	• A clear description of the safety philosophy of the system with all safeguards and safety functions clearly indicated. Sufficient design information is provided to enable the safety functions or safeguards to be assessed		P
	• Documentation for any software within the system		P
	Programming documentation is supplied in a programming design language declared by the manufacturer		P
	Safety related data and safety related segments of the operating sequence are identified and classified according to H.11.12.3		P
	There is a clear relationship between the various parts of the documentation		P
H.27.1.2.2	Class B control function		—
H.27.1.2.2.1	Design and construction requirements		—
	Software complies with software class B		P
H.27.1.2.3	Class C control function		—
H.27.1.2.3.1	Design and construction requirements		—
	Software complies with software class C		N/A
H.27.1.2.5	Circuit and construction evaluation		—
H.27.1.2.5.3	Assessment		N/A

	Only the safety related software (software class B and C) as identified according to H.27.1.2.1.2 were subjected to further assessment		N/A

TABLE R.1 / Table H.1 for software class R.1 / B – GENERAL FAULT / ERROR CONDITIONS						
Component [1]	Fault/error	Acceptable measures [2] [3] [4]	Definitions	Document reference for applied measure	Document reference for applied test	Verdict
1.CPU						—
1.1 Register	Stuck at	Functional test, or	H.2.16.5			N/A
		periodic self-test using either:	H.2.16.6	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0		P
		– static memory test, or	H.2.19.6			N/A
		– word protection with single bit redundancy	H.2.19.8.2			N/A
1.2 Void						—
1.3 Programme counter	Stuck at	Functional test, or	H.2.16.5			N/A
		periodic self-test, or	H.2.16.6	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0		P
		independent time-slot monitoring, or	H.2.18.10.4			N/A
		logical monitoring of the programme sequence	H.2.18.10.2			N/A
2. Interrupt handling and execution	No interrupt or too frequent interrupt	Functional test; or	H.2.16.5			N/A
		time-slot monitoring	H.2.18.10.4			
3. Clock	Wrong frequency (for quartz synchronized clock: harmonics/ subharmonics only)	Frequency monitoring, or	H.2.18.10.1	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0		P
		time slot monitoring	H.2.18.10.4			N/A
4. Memory						—
4.1 Invariable memory	All single bit faults	Periodic modified checksum; or	H.2.19.3.1	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0		P
		multiple checksum, or	H.2.19.3.2			N/A
		word protection with single bit redundancy	H.2.19.8.2			N/A
4.2 Variable memory	DC fault	Periodic static memory test, or	H.2.19.6	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0		P
		word protection with single bit redundancy	H.2.19.8.2			N/A

4.3 Addressing (relevant to variable and invariable memory)	Stuck at	Word protection with single bit parity including the address	H.2.19.18.2	Covered by 4.1 and 4.2	P
5.Internal data path	Communication interface				—
5.1 Data	Stuck at	Word protection with single bit redundancy	H.2.19.8.2		N/A
5.2 Addressing	Wrong address	Word protection with single bit redundancy including the address	H.2.19.8.2		N/A
6 External communication	Communication interface not safety relevant				—
6.1 Data	Hamming distance 3	Word protection with multi-bit redundancy, or CRC – single word , or	H.2.19.8.1 H.2.19.4.1		N/A
		transfer redundancy, or	H.2.18.2.2		N/A
		protocol test	H.2.18.14		N/A
6.2 Void					—
6.3 Timing	Wrong point in time	Time-slot monitoring, or scheduled transmission	H.2.18.10.4 H.2.18.18		N/A
		Time-slot and logical monitoring, or	H.2.18.10.3		N/A
		comparison of redundant communication channels by either:			—
		– reciprocal comparison	H.2.18.15		N/A
		– independent hardware comparator	H.2.18.3		N/A
	Wrong sequence	Logical monitoring, or	H.2.18.10.2		N/A
		time-slot monitoring, or	H.2.18.10.4		N/A
		scheduled transmission (same options as for wrong point in time)	H.2.18.18		N/A
7. Input/output					—
7.1 Digital I/O	Fault conditions specified in 19.11.2	Plausibility check	H.2.18.13	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0	P
7.2 Analog I/O					

| 7.2.1 A/D- and D/A- converter	Fault conditions specified in 19.11.2	Plausibility check	H.2.18.13	IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0		P
7.2.2 Analog multiplexer	Wrong addressing	Plausibility check	H.2.18.13			
8. Void						—
9 Custom chips. ASIC, GAL, Gate array	Any output outside the static and dynamic functional specification	Periodic self-test	H.2.16.6			N/A

Additional measures	Details	Reference
Stack-Memory-Overflow-Underflow-Protection	Predefined pattern at boundaries of stack memory indicate overflow or underflow in case of modification	iec60730b_dsc_stack.c IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0
Watch-Dog reset and Watch-Dog clock test	Watch-Dog reset forced by software with predefined clock, counter indicates correct timing and "reset-source register" indicates watch-dog rest	iec60730b_dsc_wdg.c IEC60730B_Library_User_Guide _56800_v4_3.pdf Revision 0
Remarks: The source code for "Stack-Test" is also included to the object code files. The source code for "Watch-DogTest" is available as open code in general.		

Supplementary information:	
The self-diagnostic routines mentioned below are foreseen for following measures of table R.1 / H.1:	
File Name	**Measure**
iec60730b_dsc_reg.c	1.1 Register
iec60730b_dsc_pc.c iec60730b_dsc_pc_object1.c iec60730b_dsc_pc_object2.c	1.3 Programme counter
iec60730b_dsc_clock.c	3. Clock
iec60730b_dsc_flash.c	4.1 Invariable memory
iec60730b_dsc_ram.c	4.2 Variable memory
iec60730b_dsc_dio.c iec60730b_dsc_dio_ext.c	7.1 Digital I/O (for output mode only)
iec60730b_dsc_aio.c	7.2.1 A/D- and D/A- converter
Remark: */*	

Photo documentation:

Examples for Emulation Boards
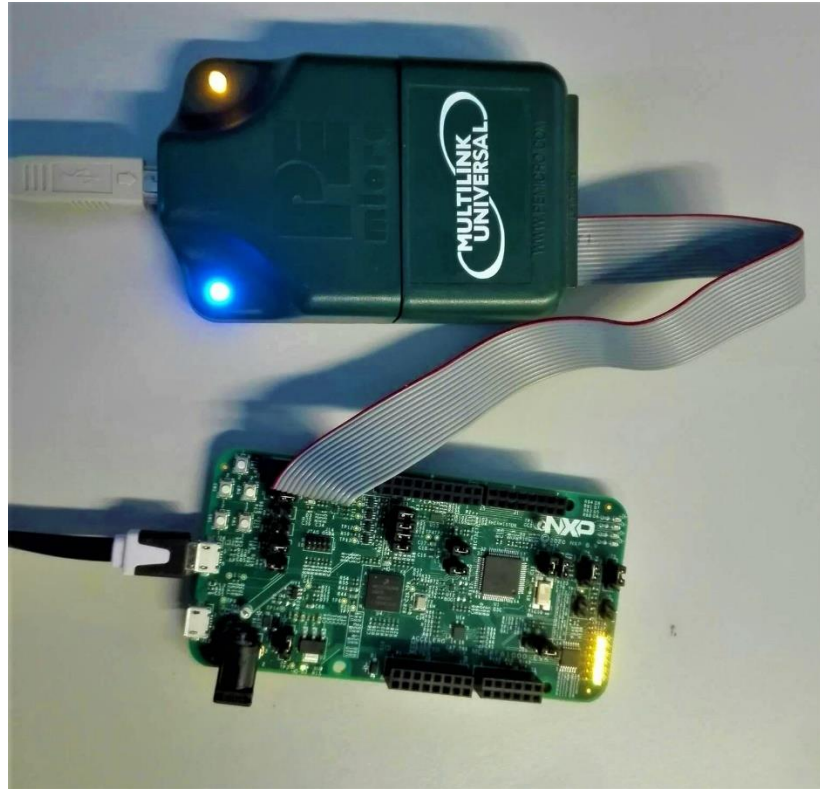
Testing and measuring equipment:

Code Warrior IDE   11.2
dsc_update   1.0.10
USB Multilink   2.0   Driver Version   12.7.0.0
Internal testing tools
PE Micro multilink
MC56F80000-EVK   development platform REV X1

End of Test Report