HOCHSCHULE RUHR WEST
UNIVERSITY OF APPLIED SCIENCES

Checklist

Fault Avoidance Software

<u>IEC 61508-3:2010:</u>

Annex A, Tables A.1 … A10

Annex B, Tables B.1 … B.9

Test Item:	Safe Temperature Monitoring
Manufacturer:	CKN GmbH
16-06-2023 Date	Unicorn Testing GmbH Name (plaintext) / Signature of tester

Modified:	05.04.19/Pa	16.06.2023/CKN							
Verified:	DS	DS							
Rev/Version:	0.1	1.0							

Content

1.1. Table A.1 – Software safety requirements specification

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1a	Semi-formal methods	Table B.7	+	+	++	++	Yes	Block Diagram of SW Modules implemented in 05_SW-System_Module_Specification	OK
1b	Formal methods	B.2.2, C.2.4	0	+	+	++	--	--	--
2	Forward traceability between the system safety requirements and the software safety requirements	C.2.11	+	+	++	++	Yes	Traceability can be verified in 00_Requirement_Tracking_Template_V01_2	OK
3	Backward traceability between the safety requirements and the perceived safety needs	C.2.11	+	+	++	++	Yes	Traceability can be verified in 00_Requirement_Tracking_Template_V01_2	OK
4	Computer-aided specification tools to support appropriate techniques/measures above	B.2.4	+	+	++	++	No	Model Based test generation is not implemented for this project	OK

NOTE 1: The software safety requirements specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application.
NOTE 2: The table reflects additional requirements for specifying the software safety requirements clearly and precisely.
NOTE 3: See Table C.1 of IEC 61508-2
NOTE 4: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

1.2. Table A.2 – Software design and development – software architecture design

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
	Architecture and design feature								
1	Fault detection	C.3.1	0	+	++	++	Yes	Fault detection mechanism implemented to detect deviation in the temperature sensor data	OK
2	Error detecting codes	C.3.2	+	+	+	++	Yes	Physical limits are verified for ensuring the data integrity of the sensor data	OK
3a	Failure assertion programming	C.3.3	+	+	+	++	No	This project has no given precondition and postcondition according to the protocol	OK
3b	Diverse monitor techniques (with independence between the monitor and the monitored function in the same computer)	C.3.4	0	+	+	0	--	--	--
3c	Diverse monitor techniques (with separation between the monitor computer and the monitored computer)	C.3.4	0	+	+	++	--	--	--
3d	Diverse redundancy, implementing the same software safety requirements specification	C.3.5	0	0	0	+	--	--	--
3e	Functionally diverse redundancy, implementing different software safety requirements specification	C.3.5	0	0	+	++	--	--	--
3f	Backward recovery	C.3.6	+	+	0	--	--	--	--
3g	Stateless software design (or limited state design)	C.2.12	0	0	+	++	--	--	--
4a	Re-try fault recovery mechanisms	C.3.7	+	+	0	0	--	--	--
4b	Graceful degradation	C.3.8	+	+	++	++	--	--	--
5	Artificial intelligence - fault correction	C.3.9	0	--	--	--	--	--	--
6	Dynamic reconfiguration	C.3.10	0	--	--	--	--	--	--
7	Modular approach	Table B.9	++	++	++	++	Yes	Sub functions are implemented as per 05_SW-System_Module_Specification	
8	Use of trusted/verified software elements (if available)	C.2.10	+	++	++	++	--	--	--

Table A.2

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
	Architecture and design feature								
9	Forward traceability between the software safety requirements specification and software architecture	C.2.11	+	+	++	++	Yes	Traceability can be verified in 00_Requirement_Tracking_Template_V01_2	OK
10	Backward traceability between the software safety requirements specification and software architecture	C.2.11	+	+	++	++	Yes	Traceability can be verified in 00_Requirement_Tracking_Template_V01_2	OK
11a	Structured diagrammatic methods **	C.2.1	++	++	++	++	Yes	Implemented as per 05_SW-System_Module_Specification	OK
11b	Semi-formal methods **	Table B.7	+	+	++	++	--	--	--
11c	Formal design and refinement methods **	B.2.2, C.2.4	0	+	+	++	--	--	--
11d	Automatic software generation	C.4.6	+	+	+	+	--	--	--
12	Computer-aided specification and design tools	B.2.4	+	+	++	++	No	Not required considering the project scope	OK
13a	Cyclic behaviour, with guaranteed maximum cycle time	C.3.11	+	++	++	++	No	Project implemented as hardware independent library	OK
13b	Time-triggered architecture	C.3.11	+	++	++	0	--	--	--
13c	Event-driven, with guaranteed maximum response time	C.3.11	+	++	++	0	--	--	--
14	Static resource allocation	C.2.6.3	0	+	++	++	No	Not required considering the project scope	OK
15	Static synchronisation of access to shared resources	C.2.6.3	0	0	+	++	No	Not required considering the project scope	OK

Table A.2

NOTE 1: Some of the methods given in Table A.2 are about design concepts, others are about how the design is represented. *NOTE 2:* The measures in this table concerning fault tolerance (control of failures) should be considered with the requirements for architecture and control of failures for the hardware of the programmable electronics in IEC 61508- 2. *NOTE 3:* See Table C.2 of IEC 61508-3 *NOTE 4:* The group 13 measures apply only to systems and software with safety timing requirements. *NOTE 5:* Measure 14. The use of dynamic objects (for example on the execution stack or on a heap) may impose requirements on both available memory and also execution time. Measure 14 does not need to be applied if a compiler is used which ensures a) that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which guarantees that in case of memory allocation error, a safe state is achieved; b) that response times meet the requirements. *NOTE 6:* Measure 4a. Re-try fault recovery is often appropriate at any SIL but a limit should be set on the number of retries. *NOTE 7:* The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.
* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application. ** Group 11, “Structured methods”. Use measure 11a only if 11b is not suited to the domain for SIL 3+4.

1.3. Table A.3 – Software design and development – support tools and programming language

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Suitable programming language	C.4.5	++	++	++	++	Yes	C programming language selected	OK
2	Strongly typed programming language	C.4.1	++	++	++	++	Yes	C programming language selected	OK
3	Language subset	C.4.2	0	0	++	++	Yes	--	OK
4a	Certified tools and certified translators	C.4.3	+	++	++	++	Yes	Eclipse IDE used for development	OK
4b	Tools and translators: increased confidence from use	C.4.4	++	++	++	++	--	--	--
NOTE 1: See Table C.3. of IEC 61508-3. *NOTE 2:* The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.									
* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.									

1.4. Table A.4 – Software design and development – detailed design

	Technique/Measure *	Siehe	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1a	Structured methods **	C.2.1	++	++	++	++	Yes	Block diagram and flow charts are used in 05_SW-System_Module_Specification	OK
1b	Semi-formal methods **	Table B.7	+	++	++	++	--	--	--
1c	Formal design and refinement methods **	B.2.2, C.2.4	0	+	+	++	--	--	--
2	Computer-aided design tools	B.3.5	+	+	++	++	No	Not required considering the project scope	OK
3	Defensive programming	C.2.5	0	+	++	++	Yes	System state initialized with failure state	OK
4	Modular approach	Table B.9	++	++	++	++	Yes	Sub functions are implemented as per 05_SW-System_Module_Specification	OK
5	Design and coding standards	C.2.6 Table B.1	+	++	++	++	Yes	MISRA C:2012 dataset used	OK
6	Structured programming	C.2.7	++	++	++	++	Yes	Implemented as per 05_SW-System_Module_Specification	OK
7	Use of trusted/verified software elements (if available)	C.2.10	+	++	++	++	--	--	--
8	Forward traceability between the software safety requirements specification and software design	C.2.11	+	+	++	++		Traceability can be verified in 00_Requirement_Tracking_Template_V01_2	OK

Anmerkung 1: See Table C.4 of IEC 61508-3.
Anmerkung 2: There is still debate about the suitability of OO software development for safety-related systems. See Annex G of IEC 61508-7 for guidance on object oriented architecture and design.
Anmerkung 3: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

** Group 1, “Structured methods”. Use measure 1a only if 1b is not suited to the domain for SIL 3+4.

1.5. Table A.5 – Software design and development – software module testing and integration

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Probabilistic testing	C,5,1	0	+	+	+	No	Not required considering the project scope	OK
2	Dynamic analysis and testing	B.6.5, Table B.2	+	++	++	++	Yes	Boundary values and negative tests implemented	OK
3	Data recording and analysis	C.5.2	++	++	++	++	No	Not required considering the project scope	OK
4	Functional and black box testing	B.5.1, B.5.2, Table B.3	++	++	++	++	Yes		OK
5	Performance testing	Table B.6	+	+	++	++	No	Not required considering the project scope	OK
6	Model based testing	C.5.27	+	+	++	++	No	Not required considering the project scope	OK
7	Interface testing	C.5.3	+	+	++	++	No	Not required considering the project scope	OK
8	Test management and automation tools	C.4.7	+	++	++	++	No	Not required considering the project scope	OK
9	Forward traceability between the software design specification and the module and integration test specifications	C.2.11	+	+	++	++	Yes	Can be verified with 06_SW-System_Test_Plan_Template_V01_0	OK
10	Formal verification	C.5.12	0	0	+	+	Yes	Implemented	OK

NOTE 1: Software module and integration testing are verification activities (see Table B.9).
NOTE 2: See Table C.5 of IEC 61508-3.
NOTE 3: Technique 9. Formal verification may reduce the amount and extent of module and integration testing required
NOTE 4: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

1.6. Tabelle A.6 – Programmable electronics integration (hardware and software)

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Functional and black box testing	B.5.1, B.5.2, Table B.3	++	++	++	++	No	Project implemented as a hardware independent library	OK
2	Performance testing	Table B.6	+	+	++	++	No	Project implemented as a hardware independent library	OK
3	Forward traceability between the system and software design requirements for hardware/software integration and the hardware/software integration test specifications	C.2.11	+	+	++	++	No	Project implemented as a hardware independent library	OK

NOTE 1: Programmable electronics integration is a verification activity (see Table A.9).
NOTE 2: See Table C.6 of IEC 61508-3.
NOTE 3: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

1.7. Tabelle A.7 – Software aspects of system safety validation

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Probabilistic testing	C.5.1	0	+	+	++	No	Not required considering the project scope	OK
2	Process simulation	C.5.18	+	+	++	++	No	Not required considering the project scope	OK
3	Modelling	Table B.5	+	+	++	++	No	Not required considering the project scope	OK
4	Functional and black-box testing	B.5.1, B.5.2, Table B.3	++	++	++	++	Yes	Implemented as per 06_SW-System_Test_Plan_Template_V01_0	OK
5	Forward traceability between the software safety requirements specification and the software safety validation plan	C.2.11	+	+	++	++	Yes	Implemented as per 06_SW-System_Test_Plan_Template_V01_0	OK
6	Backward traceability between the software safety validation plan and the software safety requirements specification	C.2.11	+	+	++	++	Yes	Implemented as per 06_SW-System_Test_Plan_Template_V01_0	OK

NOTE 1: See Table C.7 of IEC 61508-3.
NOTE 2: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

1.8. Table A.8 – Modification

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Impact analysis	C.5.23	++	++	++	++	--	--	--
2	Reverify changed software module	C.5.23	++	++	++	++	--	--	--
3	Reverify affected software modules	C.5.23	+	++	++	++	--	--	--
4a	Revalidate complete system	Table A.7	0	+	++	++	--	--	--
4b	Regression validation	C.5.25	+	++	++	++	--	--	--
5	Software configuration management	C.5.24	++	++	++	++	--	--	--
6	Data recording and analysis	C.5.2	++	++	++	++	--	--	--
7	Forward traceability between the Software safety requirements specification and the software modification plan (including reverification and revalidation)	C.2.11	+	+	++	++	--	--	--
8	Backward traceability between the software modification plan (including reverification and revalidation) and the software safety requirements specification	C.2.11	+	+	++	++	--	--	--

NOTE 1: See Table C.8 of IEC 61508-3.
NOTE 2: Techniques group 4. Impact analysis is a necessary part of regression validation. See IEC 61508-7.
NOTE 3: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

1.9. Table A.9 – Software verification

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Formal proof	C.5.12	0	+	+	++	Yes	Test results are stored in the company server	OK
2	Animation of specification and design	C.5.26	+	+	+	+	--	--	--
3	Static analysis	B.6.4, Table B.8	+	++	++	++	--	--	--
4	Dynamic analysis and testing	B.6.5, Table B.2	+	++	++	++	--	--	--
5	Forward traceability between the software design specification and the software verification (including data verification) plan	C.2.11	+	+	++	++	--	--	--
6	Backward traceability between the software verification (including data verification) plan and the software design specification	C.2.11	+	+	++	++	--	--	--
7	Offline numerical analysis	C.2.13	+	+	++	++	--	--	--
Software module testing and integration		Siehe Table A.5							
Programmable electronics integration testing		Siehe Table A.6							
Software system testing (validation)		Siehe Table A.7							

NOTE 1: For convenience all verification activities have been drawn together under this table. However, this does not place additional requirements for the dynamic testing element of verification in Table A.5 and Table A.6 which are verification activities in themselves. Nor does this table require verification testing in addition to software validation (see Table B.7), which in this standard is the demonstration of conformance to the safety requirements specification (end-end verification).
NOTE 2: Verification crosses the boundaries of IEC 61508-1, IEC 61508-2 and IEC 61508-3. Therefore, the first verification of the safety-related system is against the earlier system level specifications.
NOTE 3: In the early phases of the software safety lifecycle verification is static, for example inspection, review, formal proof. When code is produced dynamic testing becomes possible. It is the combination of both types of information that is required for verification. For example, code verification of a software module by static means includes such techniques as software inspections, walk-throughs, static analysis, formal proof. Code verification by dynamic means includes functional testing, white-box testing, statistical testing. It is the combination of both types of evidence that provides assurance that each software module satisfies its associated specification.
NOTE 4: See Table C.9 of IEC 61508-3.
NOTE 5: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

1.10. Table A.10 – Functional safety assessment

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Checklists	B.2.5	+	+	+	+	Yes	--	
2	Decision/truth tables	C.6.1	+	+	+	+	No	--	
3	Failure analysis	Table B.4	+	+	++	++	Yes	Possible failures are identified and mitigation actions implemented	OK
4	Common cause failure analysis of diverse software (if diverse software is actually used)	C.6.3	0	+	++	++	No	Diverse SW is not used	OK
5	Reliability block diagram	C.6.4	+	+	+	+	No	Not required considering the project scope	OK
6	Forward traceability between the requirements of clause 8 and the plan for software functional safety assessment	C.2.11	+	+	++	++	Yes	Implemented	OK

Anmerkung 1: See Table C.10 of IEC 61508-3
Anmerkung 2: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

2. Detailed tables

2.1. Table B.1 – Design and coding standards

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Use of coding standard to reduce likelihood of errors	C.2.6.2	++	++	++	++	Yes	Misra C:2012 standard guidelines used	OK
2	No dynamic objects	C.2.6.3	+	++	++	++	Yes	No dynamic objects implemented	OK
3a	No dynamic variables	C.2.6.3	0	+	++	++	Yes	No dynamic variables implemented	OK
3b	Online checking of the installation of dynamic variables	C.2.6.4	0	+	+	++	--	--	--
4	Limited use of interrupts	C.2.6.5	+	+	++	++	--	--	--
5	Limited use of pointers	C.2.6.6	0	+	++	++	--	--	--
6	Limited use of recursion	C.2.6.7	0	+	++	++	--	--	--
7	No unstructured control flow in programs in higher level languages	C.2.6.2	+	++	++	++	--	--	--
8	No automatic type conversion	C.2.6.2	+	++	++	++	--	--	--

NOTE 1: Measures 2, 3a and 5. The use of dynamic objects (for example on the execution stack or on a heap) may impose requirements on both available memory and also execution time. Measures 2, 3a and 5 do not need to be applied if a compiler is used which ensures a) that sufficient memory for all dynamic variables and objects will be allocated before runtime, or which guarantees that in case of memory allocation error, a safe state is achieved; b) that response times meet the requirements.
NOTE 2: See Table C.11 of IEC 61508-3
NOTE 3: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

2.2. Table B.2 – Dynamic analysis and testing

	Technique/Measure *	**Ref.**	**SIL 1**	**SIL 2**	**SIL 3**	**SIL 4**	**Verification of technique/measure**		
							Applied	**Description**	**Result**
1	Test case execution from boundary value analysis	C.5.4	+	++	++	++	Yes	Implemented	OK
2	Test case execution from error guessing	C.5.5	+	+	+	+	Yes	Implemented	OK
3	Test case execution from error seeding	C.5.6	0	+	+	+	Yes	Implemented	OK
4	Test case execution from model-based test case generation	C.5.27	+	+	++	++	No	Not required	OK
5	Performance modelling	C.5.20	+	+	+	++	No	Not required	OK
6	Equivalence classes and input partition testing	C.5.7	+	+	+	++	No	Not required	OK
7a	Structural test coverage (entry points) 100 % **	C.5.8	++	++	++	++	Yes	Implemented	OK
7b	Structural test coverage (statements) 100 %**	C.5.8	+	++	++	++	Yes	Implemented	OK
7c	Structural test coverage (branches) 100 %**	C.5.8	+	+	++	++	Yes	Implemented	OK
7d	Structural test coverage (conditions, MC/DC) 100 %**	C.5.8	+	+	+	++	Yes	Implemented	OK

NOTE 1: The analysis for the test cases is at the subsystem level and is based on the specification and/or the specification and the code.
NOTE 2: See Table C.12 of IEC 61508-3.
NOTE 3: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

** Where 100 % coverage cannot be achieved (e.g. statement coverage of defensive code), an appropriate explanation should be given.

2.3. Table B.3 – Functional and black-box testing

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Test case execution from cause consequence diagrams	B.6.6.2	0	0	+	+	No	Not required considering the project scope	OK
2	Test case execution from model-based test case generation	C.5.27	+	+	++	++	No	Not required considering the project scope	OK
3	Prototyping/animation	C.5.17	0	0	+	+	No	Not required considering the project scope	OK
4	Equivalence classes and input partition testing, including boundary value analysis	C.5.7, C.5.4	+	++	++	++	No	Not required considering the project scope	OK
5	Process simulation	C.5.18	+	+	+	+	No	Not required considering the project scope	OK

NOTE 1: The analysis for the test cases is at the software system level and is based on the specification only.
NOTE 2: The completeness of the simulation will depend upon the safety integrity level, complexity and application.
NOTE 3: See Table C.13 of IEC 61508-3.
NOTE 4: The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

2.4. Table B.4 – Failure analysis

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1a	Cause consequence diagrams	B.6.6.2	+	+	+	+	Yes	Implemented	OK
1b	Event tree analysis	B.6.6.3	+	+	+	+	--	--	--
2	Fault tree analysis	B.6.6.5	+	+	+	+	Yes	Implemented	OK
3	Software functional failure analysis	B.6.6.4	+	+	+	+	Yes	Implemented	OK

NOTE 1: Preliminary hazard analysis should have already taken place in order to categorize the software into the most appropriate safety integrity level.
NOTE 2: See Table C.14 of IEC 61508-3.
NOTE 3: The references (which are informative, not normative) "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

2.5. Table B.5 – Modelling

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Data flow diagrams	C.2.2	+	+	+	+	No	Not required	OK
2a	Finite state machines	B.2.3.2	0	+	++	++	No	Not required	OK
2b	Formal methods	B.2.2, C.2.4	0	+	+	++	Yes	Implemented	OK
2c	Time Petri nets	B.2.3.3	0	+	++	++	No	Not required	OK
3	Performance modelling	C.5.20	+	++	++	++	No	Not required	OK
4	Prototyping/animation	C.5.17	+	+	+	+	No	Not required	OK
5	Structure diagrams	C.2.3	+	+	+	++	Yes	Implemented	OK

NOTE 1: If a specific technique is not listed in the table, it should not be assumed that it is excluded from consideration. It should conform to this standard.
NOTE 2: Quantification of probabilities is not required.
NOTE 3: See Table C.15 of IEC 61508-3.
NOTE 4: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

2.6. Table B.6 – Performance testing

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Avalanche/stress testing	C.5.21	+	+	++	++	No	Not required	OK
2	Response timings and memory constraints	C.5.22	++	++	++	++	No	Not required	OK
3	Performance requirements	C.5.19	++	++	++	++	No	Not required	OK

NOTE 1: See Table C.16.
NOTE 2: The references (which are informative, not normative) “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level.

2.7. Table B.7 – Semi-formal methods

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Logic/function block diagrams	See Note 1	+	+	++	++	Yes	Block Diagram of SW Modules implemented in 05_SW-System_Module_Specification	OK
2	Data flow diagrams	See Note 1	+	+	++	++	No	Not required	OK
3	Data flow diagrams	C.2.2	+	+	+	+	No	Not required	OK
4a	Finite state machines/state transition diagrams	B.2.3.2	+	+	++	++	No	Not required	OK
4b	Time Petri nets	B.2.3.3	+	+	++	++	No	Not required	OK
5	Entity-relationship-attribute data models	B.2.4.4	+	+	+	+	No	Not required	OK
6	Message sequence charts	C.2.14	+	+	+	+	No	Not required	OK
7	Decision/truth tables	C.6.1	+	+	++	++	No	Not required	OK
8	UML	C.3.12	+	+	+	+	No	Not required	OK

NOTE 1: Logic/function block diagrams and sequence diagrams are described in IEC 61131-3.
NOTE 2: See Table C.17 of IEC 61508-3.
NOTE 3: The references “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

2.8. Table B.8 – Static analysis

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Boundary value analysis	C.5.4	+	+	++	++	No	Not required considering the project scope	OK
2	Checklists	B.2.5	+	+	+	+	No	Not required considering the project scope	OK
3	Control flow analysis	C.5.9	+	++	++	++	No	Not required considering the project scope	OK
4	Data flow analysis	C.5.10	+	++	++	++	No	Not required considering the project scope	OK
5	Error guessing	C.5.5	+	+	+	+	No	Not required considering the project scope	OK
6a	Formal inspections, including specific criteria	C.5.14	+	+	++	++	No	Not required considering the project scope	OK
6b	Walk-through (software)	C.5.15	+	+	+	+	No	Not required considering the project scope	OK
7	Symbolic execution	C.5.11	0	0	+	+	No	Not required considering the project scope	OK
8	Design review	C.5.16	++	++	++	++	No	Not required considering the project scope	OK
9	Static analysis of run time error behaviour	B.2.2, C.2.4	+	+	+	++	No	Not required considering the project scope	OK
10	Worst-case execution time analysis	C.5.20	+	+	+	+	No	Not required considering the project scope	OK

NOTE 1: See Table C.18 of IEC 61508-3.
NOTE 2: The references “B.x.x.x”, “C.x.x.x” in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. It is intended the only one of the alternate or equivalent techniques/measures should be satisfied. The choice of alternative technique should be justified in accordance with the properties, given in Annex C, desirable in the particular application.

2.9. Table B.9 – Modular approach

	Technique/Measure *	Ref.	SIL 1	SIL 2	SIL 3	SIL 4	Verification of technique/measure		
							Applied	Description	Result
1	Software module size limit	C.2.9	++	++	++	++	No	Not required considering the project scope	OK
2	Software complexity control	C.5.13	+	+	++	++	Yes	Implemented	OK
3	Information hiding/encapsulation	C.2.8	+	++	++	++	Yes	Implemented	OK
4	Parameter number limit / fixed number of subprogram parameters	C.2.9	+	+	+	+	Yes	Implemented	OK
5	One entry/one exit point in subroutines and functions	C.2.9	++	++	++	++	Yes	Implemented	OK
6	Fully defined interface	C.2.9	++	++	++	++	No	Not required considering the project scope	OK

NOTE 1: See Table C.19 of IEC 61508-3.
NOTE 2: The references "B.x.x.x", "C.x.x.x" in column 3 (Ref.) indicate detailed descriptions of techniques/measures given in Annexes B and C of IEC 61508-7.

* Appropriate techniques/measures shall be selected according to the safety integrity level. No single technique is likely to be sufficient. All appropriate techniques shall be considered.

3. History

Version	Änderung	Datum	Autor
1.0	First version	06.06.2023	CKN GmbH