Anhang N
(informativ)

Vermeiden eines systematischen Ausfalls durch den Softwareentwurf

N.1 Auswahl von Maßnahmen zur Fehlervermeidung für den Softwareentwurf

Die folgenden Tabellen enthalten Leitlinien für die Auswahl von Maßnahmen zur Fehlervermeidung für sicherheitsbezogene Embedded-Software (SRESW) oder sicherheitsbezogene Anwendungssoftware (SRASW). Tabelle 1 ist für SRASW in LVL anzuwenden, und Tabelle 2 ist für SRESW und SRASW in FVL anzuwenden.

In beiden Tabellen werden die folgenden Abkürzungen verwendet:

— r = empfohlen (en: recommended), bedeutet, dass die Anwendung dieser Maßnahme die Qualität der Softare verbessert, deren Anwendung aber nicht obligatorisch ist;

— m = obligatorisch (en: mandatory), bedeutet, dass diese Maßnahme immer angewendet werden muss

— Kanal 1 UND 2 bedeutet, dass SRESW oder SRASW in beiden Funktionskanälen der Kategorie 3 oder 4 verwendet wird;

— Kanal 1 ODER 2 bedeutet, dass SRESW oder SRASW in einem der beiden Funktionskanäle der Kategorie 3 oder 4 verwendet wird;

— vorher beurteilte Plattform bedeutet, dass die Hardware und die interne Software (SRESW) für die Sicherheitsanwendungen gestaltet und bereits beurteilt wurden, sodass sie ISO 13849 oder IEC 61508/IEC 62061 für den erforderlichen Performance Level entsprechen.

Die Maßnahmen zur Fehlervermeidung für SRESW und SRASW in Tabelle 1 und Tabelle 2 sind nach der Kategorie und dem PL eingeteilt:

a) PL a und PL b werden üblicherweise mithilfe einer Kategorie-B-Struktur und mit Software im Logikblock des Funktionskanals erreicht.

b) PL c und PL d dürfen mithilfe einer Kategorie-2-Struktur mit Software im Logikblock des Funktionskanals oder im Testeinrichtungsblock im Prüfkanal erreicht werden. Für den Prüfkanal werden die Anforderungen um einen Performance Level reduziert.

c) PL d und PL e dürfen mithilfe einer Kategorie-3-Struktur mit Software im Logikblock des Funktionskanals erreicht werden. „Kanal 1 UND 2“ bedeutet, dass Software in beiden Funktionskanälen verwendet wird. „Kanal 1 ODER 2“ bedeutet, dass Software nur in einem der beiden Funktionskanäle verwendet wird; in diesem Fall werden die Anforderungen um einen Performance Level reduziert.

d) SRASW in PL d und PL e darf außerdem mithilfe einer vorher beurteilten Plattform verwendet werden (sicherheitsbezogene Hardware in Kombination mit dem Betriebssystem und Programmierungswerkzeug). In diesem Fall wird nur eine Anwendungssoftware für beide Funktionskanäle verwendet.

Tabelle N.3 — Auswahl von Maßnahmen für SRASW in LVL

Beschreibung	**r..empfohlen**						**m..obligatorisch**					
Verwendete Kategorie	B		2				3/4				3/4[a]	
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2	Kanal 1 ODER 2	2 Kanäle	2 Kanäle
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**	**d**	**e**
Diese grundlegenden Maßnahmen sind anzuwenden:												
Entwicklungslebens-zyklus mit Verifizierungs- und Validierungstätig-keiten, siehe Bild 14 und Bild 14a	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Dokumentation der Spe-zifikation und Entwurf												
Modulare und struktu-rierte Programmierung												
Funktionsprüfungen (z. B. Black-Box-Tests)												
Geeignete Entwicklungs-aktivitäten nach Änderungen												
Die Spezifikation der sicherheitsbezogenen Software muss überprüft werden (siehe auch Anhang J) und jeder Person zur Verfügung stehen, die am Lebenszyklus des V-Modells beteiligt ist, und muss die Beschreibung enthalten von:												
Sicherheitsfunktionen mit erforderlichem PL und zugehörigen Be-triebsarten			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Leistungskriterien, z. B. Reaktionszeiten												
Hardwarearchitektur mit externen Signalschnitt-stellen												
Erkennung und Beherr-schung externer Ausfälle												

Beschreibung	**r..empfohlen**						**m..obligatorisch**					
Verwendete Kategorie	B		2				3/4				3/4[a]	
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2	Kanal 1 ODER 2	2 Kanäle	2 Kanäle
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**	**d**	**e**
Der Softwareentwurf muss folgende Merkmale haben:												
Semiformale Verfahren, um den Daten- und Kontrollfluss zu beschreiben, z. B. Zustandsdiagramm oder Programmflussdiagramm												
Modulare und strukturierte Programmierung, überwiegend realisiert durch die Bereitstellung validierter sicherheits-bezogener Funktions-block-Bibliotheken												
Funktionsblöcke mit begrenzter Codelänge												
Innerhalb des Funktions-blocks sollte die Ausfüh-rung des Codes mit einem Eingangssprung und einem Ausgangssprung erfolgen			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Architektur des Modells in drei Stufen: Ein-gänge ⇒ Verarbeitung ⇒ Ausgänge (siehe Bild 10 und Anhang J)												
Zuordnung des Sicher-heitsausgangs zu nur einem Programmteil und												
Verwendung von Techniken zur Detektion externer Ausfälle und zur defensiven Programmie-rung innerhalb von Ein-gangs-, Verarbeitungs- und Ausgangsblöcken, die zum sicheren Zustand führen												

Beschreibung	r..empfohlen						m..obligatorisch					
Verwendete Kategorie	B		2				3/4				3/4[a]	
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2	Kanal 1 ODER 2	2 Kanäle	2 Kanäle
Erforderlicher Performance Level	a	b	c	c	d	d	d	d	e	e	d	e
Softwareimplementierung/Codierung:												
Der Code muss lesbar, verständlich und test-fähig sein, und aufgrund dessen sollten symbo-lische Variablen (anstelle expliziter Hardware-adressen) angewendet werden			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Begründete oder akzep-tierte Programmierricht-linien müssen verwendet werden (siehe auch Anhang J)			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Datenintegritäts- und Plausibilitätsprüfungen (z. B. Bereichsüber-prüfungen) auf Anwen-dungsebene (defensive Programmierung) sollten verwendet werden			**r**		**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r**
Der Code sollte durch Simulation getestet werden			**r**		**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r**
Die Verifizierung sollte durch Kontroll- und Datenflussanalyse bei PL = d oder e erfolgen					**r**		**r**		**r**	**r**	**r**	**r**

Beschreibung	r..empfohlen						m..obligatorisch					
Verwendete Kategorie	B		2				3/4				3/4[a]	
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2	Kanal 1 ODER 2	2 Kanäle	2 Kanäle
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**	**d**	**e**
Prüfung:												
Das angemessene Validierungsverfahren ist der Black-Box-Test des funktionalen Verhaltens und der Leistungs-kriterien (z. B. zeitliches Leistungsverhalten)			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
I/O-Tests müssen sicher-stellen, dass die sicher-heitsbezogenen Signale in der SRASW korrekt verwendet werden			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Für PL = d oder e wird eine Testfallausführung auf der Basis von Grenz-wertanalysen empfohlen					**r**		**r**		**r**	**r**	**r**	**r**
Eine Testplanung wird empfohlen und sollte Testfälle mit Abschluss-bedingungen und erfor-derlichen Werkzeugen enthalten			**r**		**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r**
Dokumentation:												
Alle Lebenszyklus- und Änderungsaktivitäten müssen dokumentiert werden			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Die Dokumentation muss vollständig, verfügbar, lesbar und verständlich sein												
Die Codedokumentation innerhalb des Quelltextes muss Modulköpfe ent-halten mit einer juristi-schen Person, Funktions- und I/O-Beschreibung, Version der verwendeten Funktionsblock-Biblio-thek und ausreichende Kommentierung der Netzwerke/Anweisung und Deklarationszeilen												

Beschreibung	r..empfohlen						m..obligatorisch					
Verwendete Kategorie	B		2				3/4				3/4[a]	
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2	Kanal 1 ODER 2	2 Kanäle	2 Kanäle
Erforderlicher Performance Level	a	b	c	c	d	d	d	d	e	e	d	e
Konfigurationsmanagement												
Die Einführung von Verfahren und Datensicherung wird besonders empfohlen, um alle Dokumente, Softwaremodule, Ergebnisse der Verifizierung/Validierung und Werkzeugkonfiguration, die im Bezug zu einer bestimmten SRASW stehen, zu identifizieren und zu archivieren			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Änderungen												
Nach Änderungen einer SRASW muss eine Einflussanalyse zur Sicherstellung der Spezifikation durchgeführt werden. Nach Änderungen müssen angemessene Lebenszyklusaktivitäten stattfinden. Zugriffsrechte auf die Änderungen müssen geprüft und die Änderungshistorie muss dokumentiert werden. ANMERKUNG Änderungen betreffen nicht Systeme, die bereits in Betrieb sind.			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Auswahl der Werkzeuge, Bibliotheken, Sprachen:												
Für PL = e, erreicht mit einem Bauteil und dessen Werkzeug, das Werkzeug muss der einschlägigen Sicherheitsnorm entsprechen												**m**
Falls zwei verschiedene Bauteile mit unterschiedlichen Werkzeugen verwendet werden, darf der Vertrauenswert aus deren Anwendung als ausreichend angenommen werden (für PL e)									**m**			

Beschreibung	**r..empfohlen**						**m..obligatorisch**					
Verwendete Kategorie	B		2				3/4				3/4[a]	
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2	Kanal 1 ODER 2	2 Kanäle	2 Kanäle
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**	**d**	**e**
Technische Fähigkeiten, die Bedingungen erkennen, die zu systematischen Fehlern führen könnten (wie z. B. Datentyp-Unverträglich-keit, mehrdeutige dyna-mische Speicherzuord-nung, unvollständiger Aufruf von Schnittstellen, Rekursion, Zeiger-arithmetik), müssen verwendet werden			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Prüfungen sollten haupt-sächlich während der Kompilierung durchge-führt werden und nicht nur während der Laufzeit. Werkzeuge sollten Sprachenteil-mengen und Program-mierrichtlinien er-zwingen oder mindestens den Entwickler leiten oder führen.			**r**		**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r**
Wann immer angemessen durchführbar, sollten validierte Funktions-block-Bibliotheken (FB) verwendet werden – ent-weder vom Werkzeug-hersteller gelieferte sicherheitsbezogene FB-Bibliotheken oder validierte anwendungs-spezifische FB-Biblio-theken in Über-ein-stimmung mit diesem Teil der ISO 13849.			**r**		**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r**

Beschreibung	r..empfohlen						m..obligatorisch					
Verwendete Kategorie	B		2				3/4				3/4[a]	
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2	Kanal 1 ODER 2	2 Kanäle	2 Kanäle
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**	**d**	**e**
Eine begründete LVL-Teilmenge, geeignet für ein modulares Verfahren sollte verwendet werden, z. B. eine anerkannte Teilmenge der IEC 61131-3-Sprachen. Die Verwendung von graphischen Sprachen (z. B. Funktionsbaustein-Sprache, Kontaktplan) wird besonders empfohlen.			**r**		**r**	**r**	**r**	**r**	**r**	**r**	**r**	**r**
Wo SRASW und nicht-SRASW in einer Komponente kombiniert werden:												
SRASW und nicht-SRASW müssen in unterschiedlichen Funktionsblöcken codiert werden, mit sorgfältig definierten Datenschnittstellen			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Es darf keine logische Verknüpfung von nicht sicherheitsbezogenen und sicherheitsbezogenen Daten geben, die zur Herabstufung der Integrität der sicherheitsbezogenen Signale führen könnte, z. B. Verknüpfen eines sicherheitsbezogenen und eines nicht sicherheitsbezogenen Signals durch ein logisches „ODER“, dessen Ausgang sicherheitsbezogene Signale steuert.												
Verifizierung[b]												
BEISPIEL Überprüfung, Inspektion, Walk-Through oder andere geeignete Aktivitäten.			**m**		**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**

[a] Vorher beurteilte Plattform

[b] Eine Verifizierung ist nur für einen anwendungsspezifischen Code notwendig und nicht für validierte Bibliotheksfunktionen.

Tabelle N.4 — Auswahl von Maßnahmen für SRESW und/oder SRASW in FVL

Beschreibung	**r..empfohlen**						**m..obligatorisch**			
Verwendete Kategorie	**B**		**2**				**3/4**			
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2[a]	Kanal 1 ODER 2
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**
Diese grundlegenden Maßnahmen sind anzuwenden:										
Software-Sicherheits-lebenszyklus mit Verifi-zierung und Validierung, siehe Bild 14	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Dokumentation der Spezifikation und des Entwurfs (z. B. Spezifi-kation des Software-entwurfs, Spezifikation des Softwaresystem-entwurds, Spezifikation des Modulentwurfs, Codelisten einschließlich Bemerkungen)										
Modulare und struktu-rierte Programmierung (z. B. Hierarchie und Ein-schränkung der Funktio-nalität, klare Programm-struktur, Definition von Schnittstellen, gut struk-turierter Aufrufgraph, Vermeidung von Unter-brechungen, Verwen-dung von Codierungs-richtlinien)										
Beherrschung von systematischen Ausfällen (z. B. Programmablauf-überwachung, Steuerung von Fehlern im Daten-kommunikationsprozess, siehe G.2)										

Beschreibung	r..empfohlen						m..obligatorisch			
Verwendete Kategorie	**B**		**2**				**3/4**			
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2[a]	Kanal 1 ODER 2
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**
Wenn softwarebasierte Maßnahmen zur Steuerung zufälliger Hardwarefehler verwendet werden, wird die korrekte Implementierung überprüft (z. B. korrekte Durchführung von Diagnosemaßnahmen, z. B. RAM/ROM/CPU-Tests, Hardwaretests, Plausibilitätsprüfungen)	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**	**m**
Funktionsprüfungen, z. B. Black-Box-Tests (z. B. Verifizierung von korrekten Ausgabedaten basierend auf den Eingabedaten (gültig, ungültig und Grenzwerte), Kompatibilität von Schnittstellen, Zeitvorgaben)										
Geeignete Aktivitäten für den Software-Sicherheitslebenszyklus nach Änderungen (z. B. auf der Grundlage einer Einfluss-Analyse)										
Für SRESW in Bauteilen müssen die folgenden zusätzlichen Maßnahmen angewendet werden:										
Projektmanagement- und Qualitätsmanagementsystem nach z. B. IEC 61508 oder ISO 9001 (z. B. Definition des Arbeitsablaufs, der Verantwortlichkeiten, Konfigurationsmanagement, Werkzeugeinsatz)			**m**		**m**	**m**	**m**	**m**	**m**	**m**
Dokumentation aller maßgebenden Tätigkeiten während des Software-Sicherheitslebenszyklus (z. B. Dokumentation von Überprüfungen, Tests, Validierung und Verifizierung)										

Beschreibung	**r..empfohlen**						**m..obligatorisch**			
Verwendete Kategorie	**B**		**2**				**3/4**			
Teil in der Kategorie			Logik	Test-einrich-tung	Logik	Test-einrich-tung	Kanal 1 UND 2	Kanal 1 ODER 2	Kanal 1 UND 2[a]	Kanal 1 ODER 2
Erforderlicher Performance Level	**a**	**b**	**c**	**c**	**d**	**d**	**d**	**d**	**e**	**e**
Konfigurationsmanagement zur Identifizierung aller Konfigurationspunkte und -dokumente in Zusammenhang mit einer SRESW-Version (z. B. Versionskontrolle von Codelisten, Modulen, Entwurfsdokumenten, Testplänen, Freigabekontrolle, Archivierung)										
Strukturierte Spezifikation mit Sicherheitsanforderungen und Entwicklung										
Anwendung geeigneter Programmiersprachen und rechnergestützter Werkzeuge mit Vertrauen aus deren Verwendung (z. B. werden Programmierer dahingehend geschult, diese Werkzeuge mit nachgewiesener Eignung zu verwenden)			**m**		**m**	**m**	**m**	**m**	**m**	**m**
Modulare und strukturierte Programmierung, Abgrenzung von nicht sicherheitsbezogener Software, beschränkte Modulgröße mit vollständig definierten Schnittstellen, Verwendung von Entwurfs- und Codierungsrichtlinien										
Überprüfung der Codierung durch Walkthrough/Überprüfen mit Kontrollflussanalyse (zur Überprüfung auf Fehler, Qualität der Bemerkungen, Einhaltung der Codierungsrichtlinien, Klarheit, Lesbarkeit, Vollständigkeit)										