



# **Software**

## **Anforderungen und Prüfmethode**

Herausgeber und Verlag: VdS Schadenverhütung GmbH

Amsterdamer Str. 172-174

50735 Köln

Telefon: (0221) 77 66 0; Fax: (0221) 77 66 341

Copyright by VdS Schadenverhütung GmbH. Alle Rechte vorbehalten.

# Richtlinien für die Brandschutz- und Sicherungstechnik

## Software

### Anforderungen und Prüfmethoden

#### INHALT

<b>Vorbemerkung</b> .....	<b>4</b>
<b>1 Allgemeines</b> .....	<b>4</b>
1.2 Geltungsbereich .....	4
1.2 Gültigkeit.....	5
<b>2 Normative Verweisungen</b> .....	<b>5</b>
<b>3 Begriffe</b> .....	<b>5</b>
<b>4 Anforderungen und Prüfmethoden</b> .....	<b>6</b>
4.1 Allgemeines .....	6
4.2 Hauptprogrammfluss .....	7
4.3 Speicherinhaltsverzeichnis .....	8
4.4 Schnittstellen zwischen Hard- und Software .....	9
4.5 Detaillierte Programmdokumentation .....	9
4.6 Quellcode-Listing .....	10
4.7 Software-Werkzeuge.....	11
4.8 Modularität .....	11
4.9 Schnittstellen, Plausibilität .....	12
4.10 Deadlockvermeidung.....	12
4.11 Programmablaufüberwachung .....	12
4.12 Programm- und Datenspeicherung .....	13
4.13 Überwachung der Speicherinhalte.....	13
<b>Änderungen</b> .....	<b>15</b>
<b>Anhang A Fremdprodukte (Normativ)</b> .....	<b>16</b>
<b>Anhang B Versionsschema (Normativ)</b> .....	<b>17</b>
<b>Anhang C Literaturhinweise (Informativ)</b> .....	<b>18</b>

## Vorbemerkung

In Analogie zu anderen Einsatzgebieten der Technik wächst der Anteil programmierbarer elektronischer Geräte und Systeme auch im Bereich der Brandschutz- und Sicherungstechnik stetig an. Die Funktionalität dieser Geräte ist maßgeblich durch ihre Programme bestimmt, so dass deren ordnungsgemäße Ausführung und Dokumentation von besonderer Bedeutung sind.

Diese Richtlinien sind das Ergebnis einer Überarbeitung der bisherigen "Richtlinien für Gefahrenmeldeanlagen, Softwaregesteuerte Anlageteile, Ergänzende Anforderungen und Prüfmethode", VdS 2203 12/88 (01) und stellen im Wesentlichen eine Umsetzung der Anforderungen der Europäischen Normenreihe EN 54 in Prüfmethode dar. Sie beschreiben, wie die Anforderungen dieser europäischen Normenreihe abgeprüft sind, und legen Kriterien zum Nachweis der Konformität fest. Hierzu wurden die entsprechenden Anforderungen aus den vorgenannten Normen für den gesamten Geltungsbereich dieser Richtlinien sinngemäß übernommen.

Das Konzept der Anforderungen und deren Umsetzung in den hier beschriebenen Prüfmethode sieht vor, dass eine gut dokumentierte Software erkennen lässt, dass

- ein verständliches und nachvollziehbares Konstruktionsverfahren befolgt wurde,
- die Software durch kompetente Softwareingenieure korrekt gepflegt und aktualisiert werden kann, selbst wenn die ursprünglichen Konstrukteure und Planer nicht mehr verfügbar sind und
- Maßnahmen getroffen wurden, um bestimmte geläufige Ursachen für Unzuverlässigkeiten zu vermeiden oder Fehler in Zusammenhang mit der Software zu erkennen.

Die Dokumentation sollte die Software daher so beschreiben, dass Konstruktion und Programmierschritte zurückverfolgt werden können. Weiterhin sollte die Dokumentation vollständig, korrekt und konsistent sein. Es wird darauf hingewiesen, dass die Prüfung gemäß den zu Grunde liegenden Anforderungen keine Verifikation der Software ist, und nicht sicherstellen kann, dass Software oder Dokumentation fehlerfrei sind.

## 1 Allgemeines

### 1.2 Geltungsbereich

Diese Richtlinien gelten für alle Programme oder Programmkomponenten aus dem Bereich der Gefahrenmeldetechnik, die zur Verarbeitung, Darstellung und Übertragung von Nachrichten, Daten, Signalen und Informationen dienen. Für Produkte der Sicherungstechnik (z.B. Einbruchmelderzentralen) gelten sie in Ergänzung zu den entsprechenden Produktrichtlinien. Anforderungen und Prüfmethode, die nicht für Geräte der Sicherungstechnik gelten, werden explizit durch einen entsprechenden Vermerk gekennzeichnet. Gleiches gilt für Anforderungen und Prüfmethode, die gemäß der Europäischen Normenreihe EN 54 (Teile 2, 5 und 7) nur für bestimmte Geräte (z.B. Brandmelderzentralen) zugrundegelegt werden.

Die Richtlinien gelten auch für Steuereinrichtungen (z.B. für Antriebe) in der Brandschutz- und Sicherheitstechnik, Anlageteile von Zutrittskontrollanlagen und Elektronische Schlösser. Hilfsprogramme (z.B. Projektierungstools) fallen nicht in den Geltungsbereich.

## 1.2 Gültigkeit

Die Richtlinien gelten ab dem 01. März 2001; sie ersetzen die Ausgabe VdS 2203 12/88 (01). Diese kann jedoch noch für eine Übergangszeit bis zum 01. März 2002 angewendet werden.

## 2 Normative Verweisungen

Diese Richtlinien enthalten datierte und undatierte Verweise auf andere Regelwerke. Die Verweise erfolgen in den entsprechenden Abschnitten, die Titel werden im Folgenden aufgeführt. Änderungen oder Ergänzungen datierter Regelwerke gelten nur, wenn sie durch Änderung dieser Richtlinien bekannt gegeben werden. Von undatierten Regelwerken gilt die jeweils letzte Fassung.

- **EN 54–2 : 1997-10-00** Brandmeldeanlagen – Teil 2: Brandmelderzentralen
- **EN 54–2/AC : 1999-02-00** Brandmeldeanlagen – Teil 2: Brandmelderzentralen
- **prEN 54-5 : 2000-02-00** Brandmeldeanlagen – Teil 5: Wärmemelder – Punktförmige Melder
- **prEN 54-7 : 2000-02-00** Brandmeldeanlagen – Teil 7: Rauchmelder – Punktförmige Melder nach Streulicht-, Durchlicht- oder Ionisationsprinzip

## 3 Begriffe

**Anlagenbezogene Daten:** Veränderbare Daten, die zur Festlegung einer bestimmten Systemkonfiguration eines Gerätes, z.B. einer Gefahrenmelderzentrale (GMZ), benötigt werden. Diese sind z.B.:

- Zuordnung der Meldergruppen
- logische Verknüpfungen von Meldergruppen
- betriebsbedingte Zusatzfunktionen (z.B. AlarmzwischenSpeicherung, etc.)
- Zuordnung der Alarmausgänge
- Bereichszuordnung
- Funktionsfestlegung der Linien

**Daten:** Gebilde aus Zeichen oder kontinuierliche Funktionen, die auf Grund bekannter oder unterstellter Abmachungen Informationen darstellen, vorrangig zum Zweck der Verarbeitung oder als deren Ergebnis.

**Flüchtiger Speicher:** Speicherelement, welches zur Erhaltung seines Speicherinhaltes eine Energiequelle benötigt.

**Globale / Lokale Daten:** Ein in einem Modul vereinbarter und nur in diesem Modul verwendeter Name heißt lokal für dieses Modul. Tritt innerhalb eines Moduls ein Name auf, der nicht im Deklarationsteil dieses Moduls vereinbart ist, so muss der Name in einem übergeordneten Modul oder Programm vereinbart oder Name (also

ein Funktionsname) bzw. formaler Parameter des übergeordneten Moduls oder Programms sein. Solche Namen, die zudem weder Name (ein Funktionsname) noch formal Parameter dieses Moduls sind, heißen global und man spricht folglich auch von globalen Konstanten, globalen Variablen usw.

**Modul:** Ein gemäß der Logik des Programms abgegrenzter Teil, der eine bestimmte eindeutige Aufgabe wahrnimmt. Die Gliederung eines Programms in einzelne Module kann auch nach hierarchischen Gesichtspunkten erfolgen. Bei Verfahren, die aus mehreren Programmen bestehen, liegt eine modulare Struktur dann vor, wenn jedes einzelne Programm eine selbstständige, isolierte betrachtbare Funktion übernimmt.

**Nichtflüchtiger Speicher:** Speicherelement, welches zur Erhaltung seines Speicherinhaltes keine Energiequelle benötigt.

**Programm:** Nach den Regeln der verwendeten Sprache festgelegte syntaktische Einheit aus Anweisungen und Vereinbarungen, welche die zur Lösung einer Aufgabe notwendigen Elemente umfasst.

**Programmiersprache:** Eine zum Abfassen von Programmen geschaffene Sprache.

**Schnittstelle:** Gedachter oder tatsächlicher Übergang an der Grenze zwischen zwei Funktionseinheiten mit vereinbarten Regeln für die Übergabe von Daten und Signalen.

## 4 Anforderungen und Prüfmethode

### 4.1 Allgemeines

#### 4.1.1 Bereitstellung der Dokumentation

Der Hersteller muss eine Dokumentation erstellen und pflegen, die einen Überblick über die Ausführung der Software ermöglicht. Diese Dokumentation muss ausreichend detailliert sein, damit die Übereinstimmung mit den Anforderungen dieser Richtlinien geprüft werden kann.

Die Dokumentation ist - ausgenommen einzelner Dokumente, deren Weitergabe ausdrücklich der Entscheidung des Herstellers überlassen wird - gemeinsam mit dem zu prüfenden Gerät der Prüfstelle einzureichen.

In den Fällen, in denen Dokumente nicht der Prüfstelle eingereicht werden müssen, aber jederzeit zur Einsicht bereitzuhalten sind (siehe EN 54-2 : 1997-10-00), kann wie folgt verfahren werden:

- Anhand der Programmhierarchie und Gesamt-Programmstruktur (z.B. Programmablaufplan des Hauptprogramms) werden die zu prüfenden Programmmodule oder Programmpfade von der Prüfstelle benannt. Diese Unterlagen sind dann aus der verfügbaren Dokumentation zu entnehmen und kurzfristig der Prüfstelle vorzulegen.
- Alternativ hierzu kann bei besonders vertraulichen Dokumentationsteilen (z.B. Algorithmen) so verfahren werden, dass entweder der Hersteller diese Dokumentationsteile nur temporär und ggf. unter seinem Beisein zur Verfügung

stellt, oder die Prüfung dieser Dokumentationsteile beim Hersteller des Gerätes erfolgt.

- Bei Verschlussachen ist es möglich, die entsprechenden Dokumentationsteile bei der Prüfung auszuklammern.

#### **4.1.2 Form der Dokumentation**

Die Dokumentation der Software ist grundsätzlich an keine vordefinierte Form (z.B. Papierform) gebunden. Umfang, Inhalt und Aussagekraft der einzelnen Dokumente sind Gegenstand der Überprüfung. Hierbei bilden die Dokumente zusammenhängende Sätze von Informationen, aus denen die geforderten Inhalte eindeutig nachweisbar hervorgehen.

Die Dokumentation der Software sollte als Bestandteil der Gesamtdokumentation für das Gerät so aufgebaut sein, dass es möglich ist, Signale oder Informationen von der Detektion über die Verarbeitung in Hardware und Software bis zur resultierenden Aktion oder Anzeige zu verfolgen. Einschränkungen hierzu ergeben sich bei parametrierbaren Gefahrenmelderzentralen, bei denen die Informationsverarbeitung von der Parametrierung abhängt. Sie muss vollständig, klar, präzise und in sich schlüssig sein.

Alle Bestandteile der Dokumentation müssen eindeutig identifizierbar sein. Verweise auf andere Dokumente müssen ebenfalls eindeutig sein. Insbesondere muss es anhand der Dokumentation möglich sein, das Vorhandensein und die Beschaffenheit aller in diesen Richtlinien geforderten Funktionen und Eigenschaften festzustellen.

Besteht das programmierbare elektronische Gerät oder System aus mehreren programmgesteuerten Baugruppen, so sind diese jeweils separat voneinander und vollständig zu dokumentieren.

Für anwenderprogrammierte Bauelemente gelten sinngemäß die gleichen Anforderungen, sofern die Programmierung über eine den Schaltungsentwurf abstrahierende Programmiersprache vorgenommen wurde.

Für Programme und deren Dokumentation, die nicht vom Hersteller selbst oder die nicht im Auftrag des Herstellers durch Dritte erstellt wurden, gelten die im Anhang A aufgeführten Anforderungen.

Einzelne Dokumente oder auch die gesamte Dokumentation können auf elektronischen Medien verfasst und gespeichert sein. Sie sind jedoch in einer für die Prüfstelle lesbaren Form zur Verfügung zu stellen.

## **4.2 Hauptprogrammfluss**

### **4.2.1 Anforderungen**

Die Dokumentation muss bezüglich des Hauptprogrammflusses mindestens eine Funktionsbeschreibung des Hauptprogrammablaufes einschließlich

- a) einer kurzen Beschreibung der enthaltenen Module und deren Aufgabe,
- b) einer Beschreibung, aus der die Art des Zusammenwirkens, die Abhängigkeit der Module und der Objekte zueinander hervorgeht,
- c) einer Darstellung der allgemeinen Programmhierarchie,

- d) Angabe der Art, in welcher die Module aufgerufen werden, einschließlich jeglicher Unterbrechungsprozesse (z.B. Beschreibung von Software-Schnittstellen einschließlich Input- und Outputparametern)

enthalten.

Die funktionale Beschreibung des Hauptprogrammflusses muss sich klarer Methoden bedienen, die der Ausprägung der Software angemessen sind (z.B. graphische Darstellungen des Systemdesigns, des Datenflusses und des Kontrollflusses).

#### **4.2.2 Prüfung**

Die Dokumentation wird daraufhin untersucht, ob sie eine Funktionsbeschreibung des Hauptprogrammflusses enthält und ob diese die in Abschnitt 4.2.1 unter a) bis d) gelisteten Inhalte aufweist. Es wird geprüft, ob die gewählte Form der Dokumente eine klare Methodik erkennen lässt, die der Art der Software angemessen ist und die geforderten Inhalte eindeutig beschreibt.

*Anmerkung 1: Dies wird im Allgemeinen eine Darstellung der Programmstruktur in Diagrammform beinhalten (z.B. Fluss- oder State-Event-Diagramm(e)). Bei einfachen Programmen kann dies ein einzelnes Diagramm sein, im Fall von komplexeren Programmen kann es sich auch um eine hierarchische Serie von Diagrammen handeln. (Die vorgenannten Beispiele stehen exemplarisch für eine Vielzahl von anderen Darstellungsmöglichkeiten).*

*Anmerkung 2: Die in Abschnitt 4.2.1 unter a), b) und d) genannten Punkte müssen nicht separat nachweisbar sein, sondern können auch zusammen erfüllt werden. Es ist somit nicht erforderlich, dass die Beschreibung in einem separaten Dokument enthalten ist; sie kann auch in einem anderen Dokument integriert sein (z.B. in den Quellcode-Headerdateien).*

Die Dokumente/Absätze, die die geforderten Informationen enthalten, sowie die Art der Beschreibung (z.B. Diagrammart(en)) werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

### **4.3 Speicherinhaltsverzeichnis**

#### **4.3.1 Anforderungen**

Die Dokumentation muss bezüglich des Speicherinhaltsverzeichnisses mindestens eine Beschreibung enthalten, aus der hervorgeht, welche Bereiche des Speichers genutzt werden, um die Programmdateien, die anlagenbezogenen Daten und die Laufzeitdaten zu speichern.

Für den Fall, dass eine dynamische Speicherzuweisung vorgesehen ist, muss eine Trennung zwischen den Programmdateien, den anlagenbezogenen Daten und Laufzeitdaten implementiert sein, welche in Zusammenhang mit der Methode der Speicherzuweisung beschrieben sein muss.

#### **4.3.2 Prüfung**

Die Dokumentation wird daraufhin untersucht, ob sie die Speicherbereiche für die verschiedenen Zwecke wie Programmspeicherung, anlagenbezogene Daten und Laufzeitdaten beschreibt. Es muss mindestens möglich sein, den Speichertyp für Programmspeicherung, anlagenbezogene Daten sowie Laufzeitdaten zu identifizieren. Aus der Dokumentation muss nachweislich hervorgehen, dass die



anlagenbezogenen Daten von den dynamischen Daten (Laufzeitdaten) getrennt sind und auch von diesen nicht überschrieben werden können.

*Anmerkung: Dies wird häufig in Form von diagrammförmigen oder tabellarischen Memory-Maps oder über so genannte Linker/Lokater-Listings geschehen. Bei parametrierbaren Geräten (z.B. Gefahrenmelderzentralen) ist auf Grund der variablen Größe der Parametrierdaten eine Angabe der maximal möglichen Speicherbereiche für Parametrierdaten ausreichend.*

Die Dokumente/Absätze, die die geforderten Informationen enthalten, sowie die Art der Beschreibung werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

## **4.4 Schnittstellen zwischen Hard- und Software**

### **4.4.1 Anforderungen**

Die Dokumentation muss bezüglich der Schnittstellen zwischen Hard- und Software eine Beschreibung enthalten, wie die Software auf die Hardware des zu prüfenden Gerätes einwirkt. Der Hersteller muss hierzu mindestens an zwei Beispielen das Einwirken der Software auf die Hardware darlegen.

### **4.4.2 Prüfung**

Anhand der Dokumentation wird geprüft, ob beschrieben ist, wie die Software auf die Hardware des Gerätes einwirkt.

*Anmerkung: Eine Darstellung der Portbelegungen gilt als eine mögliche Form, diese Dokumentationsinhalte nachzuweisen.*

Die Dokumente/Absätze, die die geforderten Informationen enthalten, sowie die Art der Beschreibung werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

## **4.5 Detaillierte Programmdokumentation**

### **4.5.1 Anforderungen**

Die Dokumentation muss eine detaillierte Programmdokumentation enthalten. Diese muss der Prüfstelle nicht eingereicht werden. Sie ist jedoch zur Einsicht in einer Weise bereitzustellen, welche die Rechte des Herstellers auf Vertraulichkeit wahrt. In der detaillierten Programmdokumentation müssen Beschreibungen aller Programmmodule, so wie sie im Quellcode des Programms implementiert sind, mit jeweils nachstehenden Angaben vorhanden sein:

- a) Name des Moduls
- b) Datum und / oder Versionsreferenz (siehe hierzu Anhang B)
- c) eine Beschreibung der auszuführenden Aufgaben
- d) eine Beschreibung der Schnittstellen einschließlich der Art des Datentransfers, des gültigen Wertebereiches und der gültigen Daten, sofern sie nicht implizit über Prüfprozeduren der verwendeten Hochsprache gegeben ist

Ein Überblick über die gesamte System-Konfiguration, einschließlich aller Software- und Hardwarekomponenten, muss ebenfalls enthalten sein.

## 4.5.2 Prüfung

Es wird geprüft, ob eine detaillierte Dokumentation vorliegt, die die Beschreibungen aller Programmmodule enthält. Jede dieser Beschreibungen sollte mindestens die unter Abschnitt 4.5.1 a) bis d) aufgeführten Inhalte aufweisen.

*Anmerkung: Es ist nicht erforderlich, dass die Beschreibungen in einem separaten Dokument erfolgen; sie können auch in einem anderen Dokument integriert sein, z.B. in den Quellcode-Headerdateien.*

Es wird anhand mindestens drei Stichproben überprüft, ob die Beschreibungen der Module mit dem Programmfluss konsistent und hierarchisch identifizierbar sind. Dies kann durch Auswahl eines Pfades durch die Hierarchie und Verfolgung dieses Pfades durch die Modulbeschreibungen und/oder die Quellcode-Listungen geschehen.

Es wird geprüft, ob ein Überblick über das gesamte System, einschließlich aller Software- und Hardwarekomponenten anhand der Dokumentation möglich ist.

Die Dokumente/Absätze, die die geforderten Informationen enthalten, sowie die Art der Beschreibung werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

## 4.6 Quellcode-Listing

### 4.6.1 Anforderungen

Die Dokumentation muss ein Quellcode-Listing einschließlich aller globalen und lokalen Variablen, Konstanten und Labels sowie einen ausreichenden Kommentar enthalten, so dass der Programmfluss erkannt werden kann. Dieser Teil der Dokumentation muss der Prüfstelle nicht eingereicht werden. Er ist jedoch zur Einsicht in einer Weise bereitzustellen, welche die Rechte des Herstellers auf Vertraulichkeit wahrt.

### 4.6.2 Prüfung

Anhand von Stichproben wird eine Konsistenzprüfung zwischen Modulbeschreibungen und Quellcode-Listungen durchgeführt (z.B. wird überprüft, ob eine Quellcode-Listung für jedes beschriebene Modul existiert und mit der Beschreibung des jeweiligen Moduls übereinstimmt).

Es wird überprüft, ob der Quellcode für einen mit der Programmiersprache vertrauten Softwareingenieur angemessen kommentiert ist und der Programmfluss verfolgt werden kann.

*Anmerkung: Im Allgemeinen ist, um den Programmfluss erkennbar zu machen, für untere Sprachebenen (z.B. Assembler) mehr Kommentar erforderlich als für höhere Sprachebenen (z.B. C++). Die folgende Liste sollte als Hilfe für die Beurteilung einer angemessenen Kommentierung dienen, wobei die einzelnen Punkte keine harten Kriterien darstellen:*

- Routinen innerhalb eines Moduls sollten mit ihrem Zweck und den beeinflussten Variablen kommentiert sein.
- Globale Konstanten, Variablen und Einsprungmarken sollten in ihrer Bedeutung kommentiert sein, zumindest bei ihrer ersten Benennung. Die Einheiten und zulässigen Bereiche numerischer Daten sollten, wenn angebracht, beinhaltet

*sein, und der Zweck jedes Bits sollte für eine Variable, die als Bitfeld genutzt wird (z.B. um Statusflags bereitzustellen) angegeben sein.*

- *Punkte, an denen Daten Input zu oder Output von Modulen sind, sollten kommentiert sein.*
- *Steuerungsstrukturen (z.B. "Wenn-dann-Aussagen", "Fall-Unterscheidungen" usw.) und andere Entscheidungspunkte sollten mit ihrem Zweck und Ergebnis(sen) kommentiert sein, um den Programmfluss aufzuzeigen.*
- *Beginn und Ende von iterativen Schleifen sollten für die Erkennung ihres Zwecks und jeglicher Schachtelungen kommentiert sein.*
- *Das Setzen und Rücksetzen von Flags sollte kommentiert sein.*
- *Beim Assemblercode sollte jeder logische Schritt oder jede Gruppe von verbundenen Schritten, die eine identifizierbare Funktion ausführen, zweckbezogen kommentiert sein, um so den Programmfluss aufzuzeigen.*

## **4.7 Software-Werkzeuge**

### **4.7.1 Anforderungen**

Die Dokumentation muss bezüglich der Software-Werkzeuge mindestens Einzelheiten zu den bei der Programmerstellung verwendeten Software-Werkzeugen (z.B. High level design tools, Compiler, Assembler usw.) enthalten. Insbesondere die Compiler (einschließlich Version) für die Generierung des Quellcodes müssen aufgeführt sein. Die Dokumente hierzu müssen der Prüfstelle nicht eingereicht werden. Sie sind jedoch zur Einsicht in einer Weise bereitzustellen, welche die Rechte des Herstellers auf Vertraulichkeit wahrt.

### **4.7.2 Prüfung**

Es wird überprüft, ob die Dokumentation ausreichende Informationen über die Software-Werkzeuge bereithält, die für die Entwicklung dieser Software genutzt wurden.

Die Dokumente/Absätze, die die vorgenannten Informationen enthalten, werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

## **4.8 Modularität**

*Anmerkung: Dieser Abschnitt erübrigt sich, wenn Module gemäß Abschnitt 4.5 vorhanden sind.*

### **4.8.1 Anforderungen**

Um den zuverlässigen Betrieb des Gerätes sicherzustellen, muss die Software eine modulare Struktur aufweisen.

### **4.8.2 Prüfung**

Es wird anhand der Dokumentation geprüft, ob die Software eine modulare Struktur hat. Die Dokumente/Absätze, die die hierzu erforderlichen Informationen enthalten, werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

## **4.9 Schnittstellen, Plausibilität**

### **4.9.1 Anforderungen**

Um den zuverlässigen Betrieb des Gerätes sicherzustellen, müssen die Schnittstellen für manuell und automatisch generierte Daten so ausgeführt sein, dass Störungen im Programmablauf durch ungültige Daten verhindert werden.

### **4.9.2 Prüfung**

Anhand der Dokumentation wird nachvollzogen, ob die Software einen Schutz vor ungültigen Daten beinhaltet.

*Anmerkung: In der Regel wird dies über Mechanismen, wie z.B. Fehlerprüfung bei Eingabeschnittstellen, Begrenzung des Gültigkeitsbereiches, durchgeführt. Als Nachweis kann aber auch die implizite Datensicherheit auf Grund eines hoch entwickelten Sprachkonzeptes, z.B. bei Objektorientierter Programmierung, gelten.*

Die Dokumente/Absätze, die die vorgenannten Informationen enthalten, werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

## **4.10 Deadlockvermeidung**

### **4.10.1 Anforderungen**

Um den zuverlässigen Betrieb des Gerätes sicherzustellen, muss die Software so aufgebaut sein, dass das Auftreten einer Endlosschleife (Deadlock) verhindert wird.

### **4.10.2 Prüfung**

Die Software wird auf mögliche Ursachen für Deadlocks und Maßnahmen zu deren Vermeidung oder Behebung untersucht.

*Anmerkung: Anhand des Programmflusses wird stichprobenartig nachvollzogen, welche Quellen für mögliche Deadlocks (z.B. Schnittstellen mit Hardware, rekursive Routinen, iterative Schleifen, unbedingte Sprünge, die rückwärts zeigen) sich ergeben. Entsprechende Maßnahmen zur Vermeidung von Endlosschleifen oder so genannten Deadlocks werden dann anhand dieser möglichen Fehlerquellen untersucht. Mögliche Maßnahmen können sein: Time-Outs an Schnittstellen, entsprechende Softwarekonzepte auf Basis des Betriebssystems.*

Die Dokumente/Absätze, die die erforderlichen Informationen enthalten, werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

## **4.11 Programmablaufüberwachung**

Dieser Abschnitt gilt nicht für Brandmelder und Geräte der Sicherungstechnik, bei denen eine Programmablaufüberwachung in den Produktrichtlinien nicht gefordert ist.

### **4.11.1 Anforderungen**

Der Ablauf des Programms muss überwacht werden. Wenn Routinen, die im Zusammenhang mit den Hauptfunktionen des Programms stehen, nicht mehr ausgeführt werden können, muss das Gerät eine Systemstörung anzeigen bzw. wie in den Produktrichtlinien festgelegt, reagieren.

Abweichend hiervon gelten für Brandmelderzentralen bezüglich der Programmablaufüberwachung die Anforderungen der EN 54-2/AC : 1999-02-00.

#### 4.11.2 Prüfung

Es wird geprüft, ob eine Programm-Ablaufüberwachung gemäß Abschnitt 4.11.1 gegeben ist.

*Anmerkung: Normalerweise ist hier der Nachweis anhand der Dokumentation des Programms und insbesondere anhand der Stellung der Watchdogtriggerpunkte im Quellcode zu erbringen. Ein Nachweis dafür, dass eine solche Funktionalität im Gerät implementiert ist, kann auch mittels einer hardwarenahen Schaltung erbracht werden, die eine Störung im Programmablauf (jedoch nicht nur einen Fehler der Zeitbasis z. B durch Kurzschließen des Quarzes) simuliert, wobei dann die geforderte Störungsmeldung erfolgen muss.*

Die Dokumente/Absätze, die die erforderlichen Informationen enthalten, werden bestimmt und festgehalten. Alternativ hierzu kann auch die Störungsmeldung in Folge einer hardwarenah simulierten Störung geprüft werden. Die Ergebnisse werden im Prüfbericht aufgeführt.

### 4.12 Programm- und Datenspeicherung

#### 4.12.1 Anforderungen

Das Programm und die Daten (hierzu zählen Voreinstellungen wie z.B. Hersteller-Einstellungen, nicht aber Laufzeitdaten) müssen in nichtflüchtigen Speichern hinterlegt sein. Ein Beschreiben der Speicherbereiche, die das Programm und die Daten enthält, darf nur unter Benutzung spezieller Werkzeuge oder Zugangscodes und nicht während des normalen Betriebszustandes möglich sein.

Für **Brandmelder** gilt außerdem, dass anlagenbezogene Daten in Speichern hinterlegt sein müssen, die ihre Inhalte für mindestens zwei Wochen ohne externe Spannungsversorgung aufrechterhalten, es sei denn, dass Maßnahmen getroffen wurden, die das automatische Erneuern der Daten nach einem Spannungsausfall innerhalb einer Stunde gewährleisten.

Abweichend von diesen Anforderungen gelten für **Brandmelderzentralen** die Anforderungen an die Programm- und Datenspeicherung gemäß EN 54-2/AC : 1999-02-00.

#### 4.12.2 Prüfung

Es wird anhand der Dokumentation geprüft, ob die Anforderungen gemäß Abschnitt 4.12.1 an die Speicherung des Programms bzw. der Daten eingehalten werden.

Die Dokumente/Absätze, die die erforderlichen Informationen enthalten, werden bestimmt und festgehalten. Die Ergebnisse werden im Prüfbericht aufgeführt.

### 4.13 Überwachung der Speicherinhalte

Dieser Abschnitt gilt nur für Brandmelderzentralen.

#### 4.13.1 Anforderungen

Die Inhalte der Speicher, die anlagenbezogene Daten enthalten, müssen automatisch in Intervallen überprüft werden. Die Intervalle müssen den Anforderungen der Produktnorm bzw. der Produktrichtlinien entsprechen und dürfen eine Stunde nicht überschreiten. Die Überwachungseinrichtung muss bei Feststellung einer Verfälschung der Speicherinhalte eine Systemstörung signalisieren.

#### 4.13.2 Prüfung

Anhand der Programmdokumentation wird geprüft, ob die Inhalte der Speicher, die anlagenbezogene Daten enthalten, automatisch in Intervallen (entsprechend den Anforderungen), überprüft werden. Es wird geprüft, ob die Überwachungseinrichtung bei Feststellung eines verfälschten Speicherinhaltes eine Systemstörung signalisiert.

*Anmerkung: Ein alternativer Nachweis dafür, dass eine solche Funktionalität im Gerät implementiert ist, kann auch mittels einer hardwarenahen Schaltung durchgeführt werden, die einen solchen Speicherfehler simuliert. Auf Grund eines solchen Fehlers muss dann die geforderte Störungssignalisierung erfolgen.*

Die Dokumente/Absätze, die die erforderlichen Informationen enthalten werden bestimmt und festgehalten. Alternativ hierzu kann auch die Störungssignalisierung in Folge eines hardwarenah simulierten Speicherfehler geprüft werden. Die Ergebnisse werden im Prüfbericht aufgeführt.

## Änderungen

Gegenüber der Ausgabe VdS 2203 12/88 (01) wurden folgende Änderungen vorgenommen:

- Eine Angleichung der VdS-Richtlinien an andere bestehende Regelwerke, die übereinstimmende Anwendungsbereiche besitzen, wurde vorgenommen. Insbesondere die im Kapitel 13, "Zusätzliche Anforderungen an softwaregesteuerte Brandmelderzentralen" der EN 54-2 und die in der EN 54-5 bzw. EN 54-7 definierten Anforderungen liegen diesen Richtlinien zu Grunde.
- Eine Flexibilisierung der Nachweismöglichkeiten berücksichtigt die Tatsache, dass in einigen Unternehmen festgelegte Programmierstandards eingeführt sind.
- Der Anwendungsbereich wurde erweitert und ist nicht mehr nur auf Anlageteile von Gefahrenmeldeanlagen beschränkt.
- Es wird explizit ausgeführt, dass alternative Möglichkeiten zur Bereitstellung der detaillierten Programmdokumentation vorhanden sind.
- Es wird hervorgehoben, dass sämtliche Dokumente nicht der Form nach, sondern ausschließlich hinsichtlich der nachzuweisenden Inhalte überprüft werden.
- In Anhang A werden Kriterien für die Beurteilung von Fremdprodukten formuliert, die vom Gerätehersteller nur zum Einsatz gebracht werden aber nicht selbst entwickelt worden sind.
- Es werden außerdem Aussagen zur Versionsvergabe in Hinblick auf Unterscheidbarkeit und Eindeutigkeit von verschiedenen Programmversionen in Anhang B getroffen.

## **Anhang A Fremdprodukte (Normativ)**

Die folgenden Anforderungen legen Kriterien für eine Bewertung so genannter "Fremdprodukte" fest.

### **Vorbemerkung**

Unter "Fremdprodukte" sind zugekaufte Programme oder Programmkomponenten zu verstehen, die frei erhältlich sind und in verschiedenen Anwendungen zum Einsatz kommen.

Zugekaufte Programme oder Programmkomponenten kommen in Form von

- Betriebssystemen / Betriebssystemen,
- Compilern / Debuggern,
- Datenbanken,
- Treiberprogrammen, z.B. für Grafik-Systeme, LCD-Displays, sowie
- Netzwerk-Software

zum Einsatz. Sie beeinflussen wesentlich die Zuverlässigkeit der Programme des betrachteten programmierbaren elektronischen Gerätes oder Systems. Aus diesem Grund müssen Informationen über die Zuverlässigkeit der zugelieferten Produkte vorliegen.

Unter dem Begriff "Fremdprodukte" sind im Sinne dieser Richtlinien ausdrücklich keine Programme oder Programmkomponenten eingeschlossen, die im Auftrag und für den ausschließlichen Gebrauch des Herstellers des programmierbaren elektronischen Gerätes oder Systems bei einem externen Unternehmen entwickelt wurden. Diese Programmkomponenten sind ausdrücklich gemäß den vorliegenden Richtlinien zu prüfen.

### **A1 Tools, Bibliotheken**

Wenn als Teil des Entwurfs Programme oder Programmkomponenten verwendet werden sollen, die zugekauft oder zugeliefert werden, müssen diese anhand der Dokumentation eindeutig identifizierbar oder separat ausgewiesen werden.

### **A2 Eignung von Fremdprodukten, Freigabe, Händlernachweis**

Zugelieferte Programme oder zugelieferte Programmkomponenten müssen hinsichtlich Ihrer Eignung vom Hersteller geprüft und freigegeben sein.

Es muss ein Händlernachweis vorgelegt werden, aus dem hervorgeht, dass die entsprechenden Fremdprodukte frei erhältlich sind.



## **Anhang B Versionsschema (Normativ)**

Ein festes Versionsschema ist vom Hersteller vorzugeben und einzuhalten. Es muss eine Differenzierung bezüglich Umfang und Auswirkung von Modifikationen an den Programmen anhand dieses Schemas festgelegt sein. Dabei muss eine Unterscheidung verschiedener Programmversionen eindeutig möglich sein. Diese soll mindestens die meldepflichtigen und nicht meldepflichtigen Änderungen am Produkt (hier: an der Software) unterscheidbar machen.

## **Anhang C Literaturhinweise (Informativ)**

Alper, M.:

Professionale Softwaretests; Praxis der Qualitätsoptimierung kommerzieller Software  
Friedr. Viewg & Sohn Verlagsgesellschaft, Braunschweig, Wiesbaden 1994  
ISBN 3-528-05454-9

Hindel, Dr. Bernd:

Beitrag zum Technologie-Forum "Embedded Software '97"  
Qualität ist messbar: Software-Metriken  
Firma 3Soft GmbH, Erlangen

Koreimann, Dieter S.:

Lexikon der angewandten Datenverarbeitung  
de Gruyter, Berlin 1977  
ISBN 3-110-06991-1

Luck, Prof. Dr. Ing. H.; Schlossarek, U.:

Studie über softwaregesteuerte Gefahrenmeldeanlagen: Anforderungs- und Prüfkriterien der Universität Duisburg, Fachgebiet Nachrichtentechnik, Verlag VdS Schadenverhütung, Köln  
VdS 2173

Nassi, I.; Shneiderman, B.:

Flow chart techniques for structured programming  
in: Sigplan Notices 8 (1973), H. 8, S. 12-26

VDI-GIS (Hg.):

Software-Zuverlässigkeit, Grundlagen,  
konstruktive Maßnahmen, Nachweisverfahren  
VDI Verlag  
ISBN 3-528-05454-9



