Inhaltsverzeichnis

1 Grundlagen sicherheitsgerichteter Echtzeitsysteme 1
1.1 Entwicklungsstand sicherheitsgerichteter Echtzeitsysteme 2
1.2 Industrielle Prozessautomatisierungssysteme 4
1.3 Eigenschaften festverdrahteter und rechnergestützter Steuerungstechnik 8
1.4 Sicherheit und Zuverlässigkeit in der Prozessautomatisierung 10
1.5 Problemstellung sicherer rechnergestützter Prozessautomatisierung 12
1.6 Zuverlässigkeit, Sicherheit und Wirtschaftlichkeit 14
1.7 Unstetigkeit als inhärentes Risiko informationstechnischer Systeme 16
1.8 Grundlegende Konzepte des Echtzeitbetriebs 19
1.9 Einfachheit als Entwurfsprinzip 25
1.10 Sicherheitsnormen und -vorschriften 26
1.11 Ursachen und Auswirkungen von Fehlern und Ausfällen 27
1.11.1 Fehlerursachen 28
1.11.2 Hardware-Fehler 29
1.11.3 Software-Fehler 29
1.11.4 Fehlerauswirkungen 30
1.11.5 Fehlerklassifizierung 30
1.12 Allgemeines Prinzip der Fehlererkennung 32
1.12.1 Fehlererkennung durch Plausibilitätsprüfung 32
1.12.2 Fehlererkennung durch Vergleich 33
1.13 Diversität 34
1.13.1 Diversitätsarten 34
1.13.2 Fehlererkennbarkeit durch Diversität 36
1.14 Ingenieurmäßiger Entwurf sicherheitsgerichteter Echtzeitsysteme 37
Literatur 39

2 Konzepte zur sicherheitsgerichteten Prozeßautomatisierung 41
2.1 Maßnahmen zur Erzielung von Sicherheit in der Prozeßautomatisierung 41
2.1.1 Ausschluß von Fehlern und Ausfällen 42

2.1.2 Verminderung der Wahrscheinlichkeit von Fehlern und Ausfällen ... 42
2.1.3 Beeinflussung der Auswirkung von Fehlern und Ausfällen ... 42
2.1.4 Implementierungsmöglichkeiten ... 43
2.2 Festverdrahtete Elektronik in der sicherheitsgerichteten Prozeßautomatisierung ... 44
2.2.1 Regeln der Technik ... 44
2.2.2 Besonderheiten der Elektronik im Vergleich zu älteren Steuerungstechniken ... 46
2.2.3 Sicherheitskonzepte ... 48
2.2.4 Sicherheitsgerichtete verdrahtungsprogrammierbare Steuerungen ... 55
2.3 Einkanalige sicherheitsgerichtete Prozeßdatenverarbeitung ... 55
2.4 Zweikanalige sicherheitsgerichtete Prozeßdatenverarbeitung ... 56
2.4.1 Fühler und Stellglieder ... 57
2.4.2 Lichtsignale ... 57
2.4.3 Ventile ... 58
2.5 Mehrkanalige sicherheitsgerichtete Prozeßdatenverarbeitung ... 59
2.6 Systemstrukturen ... 60
2.6.1 Zweikanalige Systemstrukturen ... 61
2.6.2 Verteilte Systemstruktur ... 63
2.6.3 Struktur von Mensch-Maschine-Systemen ... 64
Literatur ... 65

3 Hardware-Systeme zur sicheren Prozeßdatenverarbeitung ... 67
3.1 Einkanalige sicherheitsgerichtete SPSen ... 68
3.1.1 Gegenüberstellung sicherheitsgerichteter VPS und SPS ... 68
3.1.2 Baumusterprüfung und Anlagenabnahme ... 69
3.1.3 Forderung der Normen an die SPS-Systemstruktur ... 70
3.1.4 Aufbau sicherheitsgerichteter SPSen ... 70
3.1.5 Tests in sicherheitsgerichteten SPSen ... 72
3.1.6 Programmierung sicherheitsgerichteter SPSen ... 74
3.2 Zweikanalige Hardware-Systeme ... 75
3.2.1 Realisierung mit Elektronik ... 75
3.2.2 Realisierung mit Mikroelektronik ... 83
Literatur ... 84

4 Zweikanalige sicherheitsgerichtete Rechnersysteme ... 85
4.1 Das System SIMIS ... 85
4.2 Das System LOGISAFE ... 86
4.3 Das System LOGISIRE ... 87
4.3.1 Maßnahmen zur Gewährleistung von Sicherheit ... 89

4.3.2 Hard- und Software-Struktur des LOGISIRE ... 90
4.3.3 Detaillierte Beschreibung der Funktionen des LOGISIRE-Betriebssystems ... 94
4.3.4 Fazit ... 96
4.4 Das System DIMI ... 96
4.4.1 Hardware-Strukturen ... 98
4.5 Erprobung des Systems DIMI ... 100
4.5.1 Ausfallarten ... 101
4.5.2 Hardware ... 102
4.5.3 Ausfallsicherheitsgerichtetes Verhalten ... 105
4.5.4 Der Vergleicher ... 105
4.5.5 Der technische Prozeß ... 107
4.5.6 Die Steuerung ... 107
4.5.7 Ergebnisse der praktischen Erprobung ... 110
Literatur ... 112

5 Entwicklung sicherheitsgerichteter Software ... 115
5.1 Systementwurf sicherheitsgerichteter Software ... 116
5.1.1 Fehlervermeidung ... 116
5.1.2 Fehlertoleranz ... 119
5.2 Qualitätssicherung von Software ... 121
5.2.1 Maßnahmen zur Software-Qualitätssicherung ... 122
5.2.2 Planung der Software-Qualitätssicherung ... 123
5.2.3 Struktur von Entwicklungsprojekten ... 125
5.2.4 Software-Anforderungsspezifikation ... 126
5.3 Ein Werkzeug zur Anforderungsspezifikation ... 127
5.3.1 Anforderungserfassung ... 127
5.3.2 Systementwurf ... 129
5.3.3 Projekt- und Konfigurationsverwaltung, Qualitätssicherung ... 130
5.3.4 Bewertung ... 131
5.4 Prinzipien des Programmentwurfs und der Programmcodierung ... 131
5.5 Software-Diversität ... 133
5.5.1 Vollständige Diversität ... 134
5.5.2 Gezielte Diversität ... 136
5.5.3 Übersetzerdiversität ... 137
5.5.4 Diversitäre Implementation ... 138
5.5.5 Diversitäre Spezifikation ... 140
5.5.6 Funktionelle Diversität ... 141
5.5.7 Zur Anwendung der Diversitätsarten ... 142
5.5.8 Mehrkanalige Software-Realisierung ... 142
Literatur ... 143

6 Software-Verifikation ... 145
6.1 Prinzipien der Software-Verifikation ... 145
6.1.1 Verifikationsplan ... 146
6.1.2 Verifikationstechniken ... 146
6.1.3 Anforderungsverifikation ... 149
6.1.4 Entwurfsverifikation ... 149
6.1.5 Modul- und Codeverifikation ... 150
6.1.6 Integrationsverifikation von Hard- und Software ... 151
6.1.7 Rechensystemvalidierung ... 151
6.2 Ausgewählte Software-Verifikationstechniken ... 152
6.2.1 Begutachtungen und Revisionen ... 152
6.2.2 Strukturiertes Nachvollziehen und Inspektionen ... 157
6.2.3 Software-Tests ... 158
6.2.4 Diversitäre Rückwärtsanalyse ... 162
6.3 Validierung von Echtzeitsystemen ... 166
6.3.1 Ereignissimulation ... 166
6.3.2 Simulation externer Umgebungen und Ausgabeverifikation ... 167
Literatur ... 172

7 Dienstgüte und Bewertung sicherheitsgerichteter Echtzeitsysteme ... 173
7.1 Leistung von Echtzeitsystemen ... 174
7.1.1 Leistungsbewertung ... 176
7.1.2 Beispiele für Benchmark-Programme ... 177
7.1.3 Laufzeitanalysatoren ... 178
7.1.4 Leistungsmonitore ... 179
7.2 Dienstqualitätskriterien von Echtzeitsystemen ... 180
7.2.1 Vorhersehbarkeit und Verlässlichkeit ... 180
7.2.2 Qualitativ-exklusive Kriterien ... 180
7.2.3 Qualitativ-graduelle Kriterien ... 182
7.2.4 Quantitative Kriterien ... 184
7.3 Bewertung sicherheitsbezogener Echtzeitsysteme ... 186
7.4 Bewertung identischer Kanäle hinsichtlich gefährlicher Ausfallarten ... 187
7.4.1 Mittlere Zeit bis zum sicherheitsbezogenen (gefährlichen) Doppelausfall ... 187
7.4.2 Beispiel: 2-aus-3-Wertungsschaltung ... 191
7.5 Bewertung identischer Kanäle hinsichtlich gefährlicher Fehlerarten ... 196
7.6 Bewertung diversitärer Kanäle hinsichtlich gefährlicher Ausfallarten ... 197
7.6.1 Beispiel: Diversitäre Implementierung der 2-aus-3-Wertungsschaltung ... 197
7.6.2 Mittlere Zeit bis zur Ausgabe sicherheitsbezogener (gefährlicher, fehlerhafter) Werte ... 199
7.6.3 Berechnung der MTDS für obiges Beispiel ... 202

7.6.4 Verbesserung der Ausfallerkennbarkeit durch Vergleich von Zwischenergebnissen ... 202
7.6.5 Bewertung von Software-Diversität ... 205
7.7 Bewertung diversitärer Kanäle hinsichtlich gefährlicher Fehlerarten ... 206
7.7.1 Ein-Bit-Vergleich ... 206
7.7.2 Beispiel: Software-Implementierung der 2-aus-3-Wertungsschaltung ... 207
7.7.3 Mehr-Bit-Vergleich ... 209
7.7.4 Beispiel: Zwei-Bit-Vergleich ... 212
7.7.5 Beispiel: Vier-Byte-Vergleich ... 215
7.7.6 Vergleich von Analogwerten ... 216
7.8 Bedeutung der Eingabewerte ... 216
Literatur ... 217

8 Das inhärent sichere Funktionsplanparadigma ... 219
8.1 Architektur und Betriebsart speicherprogrammierbarer Steuerungen ... 219
8.2 Programmiersprachen und Programmentwicklung ... 224
8.2.1 Allgemeine Merkmale der IEC-Sprachen ... 225
8.2.2 Anweisungsliste ... 228
8.2.3 Kontaktplan ... 228
8.2.4 Strukturierter Text ... 229
8.2.5 Funktionsplan ... 229
8.2.6 Sequentieller Ablaufplan ... 231
8.2.7 Anwendungsbereich höherer graphischer und textueller Sprachen ... 236
8.3 Anwendungsspezifische Programmobjekte ... 238
8.3.1 Automatisierung chemischer Prozesse ... 240
8.3.2 Notabschaltsysteme ... 241
8.4 Funktionspläne mit verifizierten Bibliotheken ... 245
8.5 Sicherheitstechnische Abnahme von Funktionsplänen ... 246
Literatur ... 250

9 Erstellung und Prüfung sicherheitsgerichteter Software ... 251
9.1 Grundlegende Methoden der Software-Qualitätssicherung ... 252
9.2 Qualitätssicherung der Dokumentation ... 253
9.3 Qualitätssicherung von Programmen ... 255
9.3.1 Inspektion von Programmen ... 256
9.3.2 Verifikation von Programmen ... 257
9.3.3 Symbolische Ausführung von Programmen ... 258
9.3.4 Test von Programmen ... 259
9.4 Industrielle Prüfung der Software von Prozeßautomatisierungssystemen ... 260

9.4.1 Grundlagen der Prüfung von Software ... 260
9.4.2 Software-Typprüfung der Funktionen von Prozeßleitsystemen ... 263
9.4.3 Automatische Dialogprüfung ... 265
9.4.4 Automatische Prüfung der Verarbeitung ... 267
9.4.5 Automatische Messung der Rechnerleistung ... 270
9.4.6 Erfahrungen ... 272
9.4.7 Weiterentwicklung ... 273
9.5 Richtlinien zur Erstellung sicherheitsgerichteter Software ... 274
9.5.1 Details von Software-Anforderungsspezifikationen ... 275
9.5.2 Entwurfsprozeduren ... 280
9.5.3 Software-Struktur ... 281
9.5.4 Selbstüberwachung ... 283
9.5.5 Entwurf und Codierung im Detail ... 285
9.5.6 Sprachabhängige Empfehlungen ... 287
9.5.7 Sprache und Übersetzer ... 288
9.5.8 Systematische Testmethoden ... 290
9.5.9 Hardware-Erwägungen ... 291
Literatur ... 291

10 Einige formale Methoden zur Programmverifikation ... 293
10.1 Analytische Verfikation mit Vor- und Nachbedingungen ... 294
10.2 Ausdrücke, Anweisungen und Beweisregeln ... 295
10.2.1 Syntax und Semantik ... 295
10.2.2 Variablen und Umgebungen ... 295
10.2.3 Auswertung von Ausdrücken ... 296
10.2.4 Ausführung einer Wertzuweisung ... 296
10.2.5 Die Null-Anweisung SKIP ... 298
10.2.6 Ausführung einer Anweisungsfolge ... 298
10.2.7 Ausführung einer IF-Anweisung ... 298
10.2.8 Ausführung einer WHILE-Schleife ... 299
10.3 Beweisregeln ... 299
10.3.1 Stärkung einer Vorbedingung, Schwächung der Nachbedingung ... 299
10.3.2 Wertzuweisungen ... 300
10.3.3 Verzweigungen ... 302
10.3.4 Anweisungsfolge ... 302
10.3.5 Schleifen ... 302
10.3.6 Beispiel: Multiplikation natürlicher Zahlen ... 305
10.3.7 Beispiel: Effiziente Multiplikation ... 305
10.4 Symbolische Ausführung von Programmen ... 309
10.4.1 Systematisierung ... 311

10.4.2 Anmerkungen zur symbolischen Ausführung 313
10.4.3 Beispiele 313
10.5 Korrektheitsbeweis eines Zeitgebers 319
10.5.1 Spezifikation 320
10.5.2 Hilfssätze 321
10.5.3 Beweis 324
10.6 Werkzeuge zur Programmverifikation 327
Literatur 329

11 Statisch und dynamisch sichere Prozessoren 331
11.1 Eine konsensual verifizierbare Prozessorarchitektur 332
11.1.1 Architektur 333
11.1.2 Datenpfade und Steuerwerk 334
11.1.3 Befehle und Operanden 336
11.1.4 Befehlsregister 337
11.1.5 Funktionale Verarbeitungseinheiten 337
11.1.6 Beispiel 337
11.1.7 Verifikation 339
11.2 Kontrollflussüberwachung 341
11.2.1 Kontrollflussfehler und -anomalien 341
11.2.2 Ansprungbefehle 341
11.2.3 Befehlsverkettung 342
11.2.4 Befehlszählerüberwachung 343
11.2.5 Erweiterte Befehlszählerüberwachung 344
11.2.6 Prozessgrößengestützte Befehlszählerüberwachung 345
11.2.7 Evaluation der Kontrollflussüberwachung 346
11.3 Datenflussüberwachung 346
11.3.1 Wertebereichskennung 348
11.3.2 Typberechtigungskennung 348
11.3.3 Einheitenkennung 349
11.3.4 Verarbeitungswegkennung 350
11.3.5 Zeitschrittkennung 352
11.3.6 Fristkennung 352
11.3.7 Zykluszeitkennung 353
11.3.8 Signaturkennung 354
11.3.9 Übersicht über alle Datenelementkennungen 355
11.3.10 Evaluation der Datenspezifikationsarchitektur 355
Literatur 356

12 Eine funktionsplanabbildende Prozeßrechnerarchitektur 359
12.1 Anforderungen an die Rechnerarchitektur 360
12.2 Informationsverarbeitung 363

12.2.1 Abbildung natürlicher Systemstrukturen in der Rechnerarchitektur ... 363
12.2.2 Erweiterbarkeit von Rechenanlagen ... 366
12.2.3 Parallelität zur Erhöhung der Rechenleistung ... 367
12.2.4 Modularität und Parallelität ... 374
12.2.5 Kommunikation ... 375
12.2.6 Aspekte des Zeitverhaltens ... 376
12.2.7 Betriebssystem ... 377
12.2.8 Regeln zum Anpassen eines Rechners an ein automatisierungstechnisches Problem ... 379
12.2.9 Software-Konzeption ... 381
12.3 Schnittstellen zu Sensoren und Aktoren ... 382
12.3.1 Anwendungsspezifische gegenüber Standardmodulen ... 382
12.3.2 Sensor-/Aktoranbindung auf der Signalebene ... 384
12.3.3 Erweiterbarkeit und Sensor-/Aktoranbindung auf der Busebene ... 391
12.3.4 Sensor-/Aktoranbindung auf der Prozessorebene ... 393
12.3.5 Feldbusse und sensorlokale Prozessoren ... 394
12.4 Ein Einsatzbeispiel aus der Robotik ... 394
12.4.1 Gesamtstruktur des Steuerungssystems der Karlsruher Hand ... 395
12.4.2 Hardware-Komponenten der Rechenanlage der Karlsruher Hand ... 398
Literatur ... 400

13 Fallstudien sicherheitsgerichteter programmierbarer elektronischer Systeme ... 401
13.1 Ein leicht verifizierbares ausfallsicherheitsgerichtetes PES ... 402
13.1.1 Fuzzy-Logik als Entwurfsprinzip eines sicherheitsgerichteten PES ... 402
13.1.2 Ursache-/Wirkungstabellen ... 403
13.1.3 Eine auf Fuzzy-Logik beruhende programmierbare Steuerung ... 404
13.1.4 Sicherheitsaspekte ... 409
13.2 Architektur einer sicherheitstechnisch abnehmbaren SPS ... 409
13.2.1 Hardware-Architektur ... 410
13.2.2 Software-Verifikation ... 418
13.2.3 Einige Anmerkungen ... 423
13.3 Eine anwendungsorientierte asymmetrische Mehrprozessorarchitektur ... 424
13.3.1 Das Architekturkonzept ... 426
13.3.2 Die Ereigniserfassungsschicht ... 427

13.3.3 Die Primärreaktionsschicht ... 429
13.3.4 Die Sekundärreaktionsschicht ... 429
13.3.5 Bewertung ... 430
13.3.6 Anwendungen ... 430
13.4 Zeitgenau arbeitende Prozeßperipherie ... 432
13.4.1 Notwendige Funktionen und ihr Aufruf in PEARL ... 433
13.4.2 Implementierung der Hardware-Unterstützung ... 434
13.4.3 Betriebssystemunterstützung ... 436
Literatur ... 437

14 Zeitsignalverbreitung und rechnerinterne Zeitverwaltung ... 439
14.1 Terrestrische Zeitsignalverbreitung ... 440
14.1.1 Zeitübertragung ... 440
14.1.2 Empfang von Zeitsignalen ... 442
14.1.3 DCF 77 ... 445
14.1.4 Verfügbare Funkuhren ... 448
14.2 Zeitsignalverbreitung mit GPS ... 448
14.2.1 Überblick über GPS ... 449
14.2.2 GPS-Empfänger ... 451
14.2.3 Fehlerquellen ... 456
14.3 Hochgenaue simultane Ereignisverarbeitung ... 459
14.3.1 Prozessor zur Bearbeitung gleichzeitiger Ereignisse ... 460
14.3.2 Hochgenaue Zeitsteuereinheit ... 462
14.3.3 Formale Funktionsbeschreibung des Ereignisprozessors ... 467
14.3.4 Algorithmen der Task-Verwaltung ... 471
14.3.5 Bewertung ... 474
Literatur ... 476

15 Unterbrechungsfreie asynchrone Echtzeitverarbeitung mit Zustandswiederherstellung zur Laufzeit ... 477
15.1 Rechenprozeßverarbeitung ohne asynchrone Unterbrechungen ... 478
15.1.1 Paradigmen des Echtzeitbetriebs ... 478
15.1.2 Entwurfsprobleme ... 486
15.1.3 Lösungskonzept ... 489
15.1.4 Ausführungsbeispiel ... 504
15.2 Neuaufsetzen im laufenden Betrieb redundant arbeitender Prozessoren ... 517
15.2.1 Neuaufsetzen von Echtzeitsystemen ... 518
15.2.2 Gerätetechnische Sortierung von Datenwörtern nach Altersklassen ... 528
15.2.3 Beispielhafter Aufbau der Schaltungskomponenten ... 538
Literatur ... 549

16 Ein sicherheitsgerichteter Feldbus 551
16.1 Feldbussysteme 552
16.2 Signalcodierung 555
16.2.1 Fehlerreduktion durch Signalcodierung 555
16.2.2 Signalcodierung mit modifizierter Frequenzumtastung 555
16.2.3 Detektierung signalcodierter Datenbits 560
16.2.4 Detektierung von Synchron- und Statussignalen 569
16.2.5 Signalangepasstes Filter 570
16.2.6 Ein- und Ausgangsstufen 573
16.3 Datencodierung 575
16.3.1 Datencodierung von Nibbles 576
16.3.2 Berechnung der Korrekturstellen 577
16.3.3 Hamming-Codierung von Daten-Nibbles 580
16.3.4 Decodierung gesicherter Daten-Nibbles 582
16.3.5 Fehlerwahrscheinlichkeiten 584
16.3.6 Vergleich mit marktüblichen Feldbussystemen 590
16.4 Zeitsynchronisierung auf Ringbussen 592
16.4.1 Zeitmessung auf einem Doppelringbus 592
16.4.2 Zeitmessung auf einem Einzelringbus 596
16.4.3 Bearbeitung zeitabhängiger Aufträge 600
16.4.4 Synchronisierung der Slave-Bausteine 601
16.5 Doppelringbus 604
16.6 Summenrahmentelegramm 608
16.6.1 Signalcodierung bestimmter Telegrammabschnitte 609
16.6.2 Aufbau der Slave-Bausteine 609
16.6.3 Synchronisation der Slave-Bausteine 611
16.6.4 Prioritätssteuerung der Slave-Bausteine 612
16.6.5 Übertragungszeit des Summenrahmentelegramms 613
16.6.6 Vergleich mit marktüblichen Feldbussystemen 614
Literatur 619

17 Sicherheitsgerichete Echtzeitprogrammierung in PEARL 621
17.1 Echtzeitprogrammiersprache PEARL 622
17.1.1 Entstehungsgeschichte und Eigenschaften 622
17.1.2 Übersicht über die wichtigsten Spracheigenschaften 624
17.1.3 Programmstruktur 626
17.1.4 Algorithmik 628
17.1.5 Prozessdatenein- und -ausgabe 632
17.1.6 Echtzeitprogrammierung 639
17.2 PEARL für verteilte Systeme 643
17.3 Sicherheitsgerichtetes PEARL 650
17.3.1 Verifizierbarkeitsorientierte Sprachteilmengen 650

17.3.2 SIL1: Konstruktiver Ausschluss vieler Fehlerquellen 652
17.3.3 SIL2: Vorhersehbares Zeitverhalten 653
17.3.4 SIL3: Funktionspläne 655
17.3.5 SIL4: Ursache-/Wirkungstabellen 656
17.3.6 Zusammenfassende Darstellung der Sprachteilmengen 658
17.4 Sichere Ablaufpläne 659
17.5 PEARL als Spezifikationssprache 661
Literatur 666

18 Ablaufplanung und Zuteilbarkeitsanalyse für den Mehrprozessbetrieb 667
18.1 Graphisches Programmieren im Großen 668
18.2 Prinzipien der Ablauforganisation 669
18.3 Zeitsynchrone Zuteilung 670
18.4 Multitasking 672
18.5 Terminbezogene dynamische Prozessorzuteilung 678
18.5.1 Struktureigenschaften des Antwortzeitalgorithmus 682
18.5.2 Hinreichende Bedingungen der zeitgerechten Ausführbarkeit unter Beachtung von Betriebsmittelreservierungen 684
18.5.3 Nichtpräemptive Antwortzeitzuteilung 687
18.5.4 Vermeidung von Kontextumschaltungen ohne Verletzung zeitgerechter Ausführbarkeit 688
18.5.5 Überlastvermeidung durch lastadaptive dynamische Zuteilung 691
18.6 Statische Zuteilbarkeitsanalyse 692
18.6.1 Zuteilbarkeitsanalysierbare Echtzeitprogrammiersprachen 694
18.6.2 Ein Zuteilbarkeitsanalysator 696
18.6.3 Bewertung 702
Literatur 702

Sachverzeichnis 705