

Xpert.press

Softwareentwicklung eingebetteter Systeme

Grundlagen, Modellierung, Qualitätssicherung

Bearbeitet von
Peter Scholz

1. Auflage 2005. Buch. xii, 232 S. Hardcover
ISBN 978 3 540 23405 0
Format (B x L): 15,5 x 23,5 cm
Gewicht: 537 g

[Weitere Fachgebiete > EDV, Informatik > Hardwaretechnische Grundlagen > Systemverwaltung & Management](#)

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei


DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

1 Einleitung

Das erste Kapitel enthält eine allgemeine Hinführung zum Thema. Diese beginnt mit einer kurzen Motivation und geht dann nach einer Klassifikation eingebetteter Systeme auf relevante Beispiele und betroffene Branchen sowie Anwendungen ein. Es folgt ein Überblick über die Struktur eingebetteter Systeme. Nach einer Reihe begrifflicher Definitionen schließt das Kapitel mit einer Diskussion zentraler Herausforderungen auf diesem Gebiet. Nach der Lektüre dieses Kapitels sollte der Leser die zentrale Rolle eingebetteter Systeme für die Informatik verstanden haben, sie von anderen Systemen unterscheiden können und die wichtigsten Anwendungen und Begriffe kennen, sowie die noch zu lösenden Herausforderungen auf diesem Gebiet einschätzen können.

Kapitelübersicht

1.1 Motivation

Unter einem *eingebetteten System* verstehen wir ein in ein umgebendes technisches System eingebettetes und mit diesem in Wechselwirkung stehendes Computersystem. Es übernimmt dort komplexe Steuerungs-, Regelungs-, Überwachungs- und Datenverarbeitungsaufgaben und verleiht damit dem umgebenden System oft einen entscheidenden Wettbewerbsvorsprung. Solche Systeme sind heute auf breiter Front in alle Bereiche der Technik eingedrungen. Mehr als *neun von zehn* aller elektronischen Bauelemente sind in eingebetteten Systeme implementiert. In der Tat sind in modernen Personenkraftwagen der Oberklasse zwischen 70 und 80 integrierte und miteinander vernetzte Steuergeräte enthalten (Broy et al. 1998).

Eingebettete Systeme

Reaktive Systeme nehmen heute in modernen Computersystemen eine bedeutende Rolle ein. Im Gegensatz zu rein transformationellen Systemen, also solchen, die lediglich Eingaben, die beim Systemstart vollständig zur Verfügung stehen, in Ausgaben, die erst bei der

Reaktive Systeme

Systemterminierung vollständig berechnet sind, verarbeiten, interagieren reaktive Systeme beständig mit ihrer Umgebung (Halbwachs, 1993), (Harel, Pnueli, 1985). Dabei wird die Interaktion des Systems mit seiner Umgebung weniger durch das System selbst getrieben, sondern von Ereignissen aus der Umgebung.

Reaktive Systeme finden ihre Anwendung in der Flugzeug-, Automobil- oder Telekommunikationselektronik. Bemerkenswerterweise sind bereits mehr Mikroprozessoren in reaktiven Systemen als in Personalcomputern eingebaut. So hat eine statistische Erhebung im Jahre 2002 ergeben, dass von 8,3 Milliarden weltweit produzierten Prozessoren 8,15 Milliarden in eingebetteten Systemen verbaut wurden, aber lediglich 150 Millionen in transformationellen Computersystemen (wie Personal Computer, Server, Mainframes usw.). Dies entspricht einem Verhältnis von 98:2 zugunsten der eingebetteten Systeme. In (Fränzle, 2002) ist dieser Zusammenhang aus Sicht eines eingebetteten Systems humoristisch und doch sehr treffend formuliert: „Ach wie gut, dass niemand daran denkt, dass mich ein Computer lenkt“.

Eingebettete Systeme sind eine der schnellstwachsenden Branchen unter den Informatikanwendungen. In Zukunft darf sogar eher noch mit einer Zunahme dieses Ungleichgewichts gerechnet werden. Es zeigt auch deutlich auf, wo in Zukunft interessante Aufgabengebiete und berufliche Chancen für Informatiker und Informationstechniker liegen werden.

Die Hauptgründe für das große Interesse an eingebetteten Systemen sind Fortschritte in Schlüsseltechnologien wie Mikroelektronik und formalen Methoden zu ihrer exakten Beschreibung als Grundlage für Sicherheit und Zuverlässigkeit sowie die sich daraus ergebende Vielfalt von Anwendungen.

Aktuelle Trends

Aktuelle Trends beim Entwurf eingebetteter Systeme sind:

- Steigender Anteil des elektronischen Teilsystems, dabei steigender Anteil des digitalen Teilsystems sowie *steigender SW-Anteil*,
- Trend zu immer mehr Intelligenz und fortschreitender *Vernetzung*.
- *Entwurfskompromiss*: kostengünstige Standardkomponenten vs. schnelle Spezialhardware



1.2 Klassifikation, Charakteristika

Die heute verfügbaren Computersysteme können in drei unterschiedliche Klassen eingeteilt werden (Halbwachs, 1993), (Harel, Pnueli, 1985): (rein) transformationelle, interaktive und reaktive Systeme. Sie werden in erster Linie durch die Art und Weise unterschieden, wie sie Eingaben in Ausgaben transformieren (Scholz, 1998).

Transformationelle Systeme transformieren nur solche Eingaben in Ausgaben, die zum Beginn der Systemverarbeitung vollständig vorliegen. Die Ausgaben sind dann nicht verfügbar, bevor die Verarbeitung terminiert. In solchen Systemen ist der Benutzer, oder allgemeiner, die Systemumgebung, nicht in der Lage, während der Verarbeitung mit dem System selbst zu interagieren und so Einfluss auf ihr Ergebnis zu nehmen.

Transformationelle Systeme

Interaktive Systeme dagegen, also beispielsweise Betriebssysteme, erzeugen Ausgaben nicht nur erst dann, wenn sie terminieren, sondern sie interagieren und synchronisieren stetig mit ihrer Umgebung. Diese Interaktion wird durch das System selbst und nicht etwa durch seine Umgebung bestimmt: Wann immer das System neue Eingaben zur Fortführung weiterer Verarbeitungsschritte benötigt, wird die Umgebung bzw. der Benutzer hierzu aufgefordert (engl. to prompt) – das System synchronisiert sich auf diese Weise *proaktiv* mit seiner Umgebung. Wird diese Synchronisierung dagegen durch die Systemumgebung anstelle des Systems bestimmt, so nimmt das System selbst eine reaktive Rolle ein und wir sprechen von einem reaktiven System.

Interaktive Systeme

Ein bedeutender Unterschied zwischen den beiden Arten von Systemen ist der „Herr“ der Interaktion. Bei interaktiven Systemen ist es der Computer. Die „Anfragenden“ warten bis sie bedient werden, der Computer entscheidet wer, wann und wie behandelt wird. Nennenswerte Herausforderungen bei interaktiven Systemen sind die Vermeidung von Verklemmungen (engl. deadlocks), „Fairness“ und die Konsistenz verteilter Informationen. Bei reaktiven Systemen ist es dagegen die Umgebung, die vorschreibt was zu tun ist. Der Computer hat in der vorgegebenen Zeit auf Stimuli der Umgebung zu reagieren. Größte Belange sind die Sicherheit und die Rechtzeitigkeit (Berry, 1998).

Der Unterschied zwischen reaktiven und interaktiven Systemen hat großen Einfluss auf den Verhaltensdeterminismus in diesen Systemen. Interaktive Systeme werden größtenteils als nichtdeterministisch angesehen, denn der Computer trifft intern die Entscheidung ob und wann eine Anfrage beantwortet werden soll. Auch die Reak-

tion auf eine Sequenz der Anfragen muss nicht immer gleich sein. Im Gegensatz dazu ist das deterministische Verhalten ein fester Bestandteil eines reaktiven Systems. Die Reaktion muss eindeutig durch die Eingabesignale und evtl. deren zeitliche Reihenfolge definiert sein. Beispiel dafür ist die Steuerung eines Flugzeugs oder Autos.

Das Verhalten eines nichtdeterministischen Systems ist weitaus komplexer zu modellieren als das eines deterministischen. Die Charakteristik der beiden Systeme muss unbedingt bei der Entwicklung von geeigneten Techniken, Methoden und Werkzeugen berücksichtigt werden.

Reaktive Systeme

Die wichtigsten Eigenschaften *reaktiver Systeme* kann man wie folgt charakterisieren (Halbwachs, 1993): Sie arbeiten oftmals nebenläufig, müssen äußerst zuverlässig sein und dabei Zeitschranken einhalten. Sie können sowohl in Hardware wie auch in Software realisiert und auf einer komplexen, verteilten Systemplattform implementiert werden.

Die funktionale Korrektheit reaktiver Systeme spielt eine entscheidende Rolle bei der Entwicklung solcher Systeme. Nicht nur, dass sich die Markteinführung eines Produktes verzögern kann, wenn Systemfehler erst in einer späten Entwicklungsphase entdeckt werden, schlimmer noch sind kostspielige Rückrufaktionen, wenn Fehler erst dann bekannt werden, wenn das Produkt schon beim Endabnehmer angekommen ist. Schließlich können unentdeckte Fehler bei reaktiven Systemen in kritischen Anwendungen auch noch zu Konsequenzen von größerer Tragweite führen.

In aller Regel sind reaktive Systeme in eine komplexe, beispielsweise mechanische, chemische oder biologische Systemumgebung eingebettet. Es handelt sich dann um *eingebettete Systeme*. Mit Umgebung bezeichnen wir nicht nur die natürliche Umgebung in der das kontrollierende, reaktive System agiert, sondern auch den kontrollierten Teil des komplexen Gesamtsystems. Für solche reaktive, eingebettete Systeme ist in aller Regel nicht nur deren funktionale Korrektheit wichtig, sondern auch ihre Reaktionszeiten. Man spricht in diesem Zusammenhang von *Echtzeitsystemen*, vgl. Abbildung 1.1 (siehe auch Abschnitt 4).

Eingebettete Systeme lassen sich nach einer Reihe weiterer, unterschiedlicher Kriterien klassifizieren. Zum einen bietet sich eine Einordnung in die im folgenden Abschnitt 1.3 genannten Produktkategorien an. Zum anderen treten diese Systeme in unterschiedlichen technischen Ausprägungen auf. So lassen sich *kontinuierliche* von *diskreten* und *verteilte* von *monolithischen* Systemen unterscheiden. Enthält ein System sowohl kontinuierliches als auch

diskretes Verhalten, so spricht man von einem *hybriden* System (siehe Abschnitte 1.4 und 5.8).

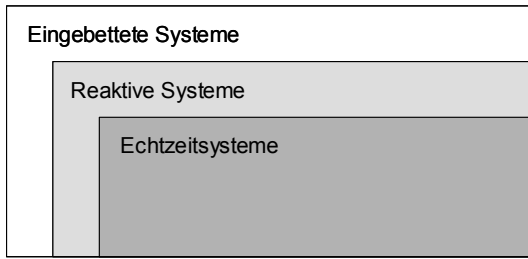


Abb. 1.1
Klassifikation
eingebetteter
Systeme

Eine weitere Zuordnungsmöglichkeit ergibt sich aufgrund der Sicherheitsrelevanz der Aufgaben, die ein eingebettetes System im Gesamtsystem übernimmt. Dies führt zur Einteilung in sicherheitskritische und nicht-sicherheitskritische eingebettete Systeme. Ein *sicherheitskritisches eingebettetes System* liegt vor, wenn von seiner korrekten Funktionsweise Menschenleben oder die Unversehrtheit von Einrichtungen abhängen. Während in der Konsumelektronik hauptsächlich nicht sicherheitskritische eingebettete Systeme zum Einsatz kommen, treten in der Avionik, Medizintechnik, insbesondere aber auch im Kraftfahrzeugbereich zunehmend sicherheitskritische eingebettete Systeme auf. Darüber hinaus lassen sich zeitkritische Systeme von nicht-zeitkritischen Systemen differenzieren. Bei letzteren spielt die Zeitspanne, innerhalb derer das System auf Eingabesignale reagiert, keine Rolle.

*Sicherheits-
kritische
Systeme*

Damit zeichnet sich gerade die Automobiltechnik gegenüber den anderen später (siehe Abschnitt 1.3) genannten Produktkategorien durch zwei wesentliche Aspekte aus:

- Kraftfahrzeuge und damit die in ihnen (in Form von Steuergeräten) auftretenden eingebetteten Systeme bilden einen Massenmarkt.
- Eingebettete Systeme in Kraftfahrzeugen übernehmen zeit- und zunehmend auch sicherheitskritische Aufgaben.

Wir wollen uns daher bei unseren Betrachtungen im Folgenden vor allem auf eingebettete Systeme in der Automobiltechnik konzentrieren.

Eingebettete Systeme müssen zu jedem Zeitpunkt ein *deterministisches*, also vorhersagbares Verhalten besitzen. Diese Eigenschaft gilt für ihre Realisierung, jedoch nicht notwendigerweise für ihre Anforderungs- oder Entwurfsspezifikation. In den Entwicklungsphasen der Anforderungsanalyse (engl. requirements analysis) bzw.

*(Nicht-)
Determinismus*

des Entwurfs (engl. design) kann die Spezifikation eines eingebetteten Systems durchaus Nichtdeterminismen aufweisen. Hier handelt es sich im Falle der Verhaltensspezifikation um das semantische Ergebnis gewünschter Unter- bzw. Überspezifikationen. Mit beiden Spezifikationsmitteln wird der Entwickler in die Lage versetzt, Teile des Systemverhaltens noch offen zu lassen, d. h. davon zu abstrahieren. Eine Konkretisierung der Spezifikation und damit Auflösung der Nichtdeterminismen muss zu einem späteren Zeitpunkt, spätestens jedoch zur Implementierung des Systems geschehen. Eine an das geschilderte Vorgehen angepasste Entwurfsmethode ist die sogenannte schrittweise Verfeinerung bzw. der inkrementelle Entwurf.

1.3 Anwendungen, Beispiele und Branchen

Anwendungs- gebiete eingebetteter Systeme

Eingebettete Systeme sind heute wie bereits eingangs erwähnt und mit Zahlenmaterial unterlegt schon weit verbreitet. Sie sind insbesondere in folgenden Anwendungsgebieten zu finden:

- Telekommunikation (Vermittlungsanlagen, Mobiltelefone, Telefone, Faxgeräte etc.)
- Haushalt (Waschmaschine, Mikrowelle, Fernseher etc.)
- Geräte für Freizeit und Hobby
- Automobiltechnik (ABS, Wegfahrsperr, Navigationssysteme, Verkehrsleitsysteme etc.)
- Öffentlicher Verkehr (Fahrkartenautomat, etc.)
- Schienenverkehr (ICE, TGV)
- Luft- und Raumfahrttechnik, Avionik (Airbus-Reihe, Boeing 777, Militärsektor)
- Fertigungstechnik, Anlagenbau
- Steuerungs- und Regelungstechnik
- Medizintechnik
- Umwelttechnik
- Militärtechnik
- u. v. m.

Besonders interessante Anwendungsgebiete eingebetteter Systeme sind die *Avionik* (Flugzeugbau) und der *Automobilbau*. In einem modernen Fahrzeug sind heute oftmals mehr als 70 Mikroprozessoren oder elektronische Kontrolleinheiten integriert, die wir unter dem Begriff Steuergeräte zusammenfassen. Die Zahl der Prozessoren ist in den letzten Jahren sprunghaft gestiegen und wird auch in Zukunft noch weiterhin beständig anwachsen. Auf diesen Mikroprozessoren ist eine immer komplexer werdende Software implementiert, die im Auto Steuerungsfunktionen angefangen von der Zentralverriegelung über die Klimaanlage bis hin zur Motorsteuerung übernimmt. Die so entstehenden Kontrolleinheiten unterstützen den Fahrer in Standard- oder auch sicherheitskritischen Situationen.

Ein konkretes Beispiel aus dem Automobilbau ist dabei folgendes: Nahezu alle heute gebauten Kraftfahrzeuge sind mit einem *Airbag-System* ausgestattet. Der Airbag bläst sich im Falle eines Unfalls mit hoher Geschwindigkeit auf und schützt so den Autoinsassen vor dem Aufprall auf das Lenkrad bzw. den Armaturenräger.

Beispiel: Airbag

Das korrespondierende eingebettete System arbeitet in etwa wie folgt: Im Fahrzeugs sind mehrere Sensoren verbaut, die dessen Beschleunigung messen. Prallt ein Wagen auf ein Hindernis, entsteht eine negative Beschleunigung. Diese wird vom Sensor registriert, der einen Gasgenerator zündet und damit das Aufblasen des Airbags in Gang setzt. Der geschilderte Vorgang passiert in der sehr kurzen Zeit von nur wenigen Hundertstel Sekunden. Die hohe Aufblasgeschwindigkeit des Airbags stellt sicher, dass das Luftkissen bereits voll aufgeblasen ist, wenn der Oberkörper des Autoinsassen nach vorne geschleudert wird. Nach dem selben Prinzip funktionieren auch zusätzliche Airbags für den Seiten-, Brust-, oder Kopfbereich. Hier sind die Sensoren an den jeweiligen Stellen an den Seiten des Wagens angebracht.

Eingebettete Steuer- und Regelsysteme sind mittlerweile aber nicht nur in Kraftfahrzeugen und Flugzeugen verbaut, sondern in beinahe allen technischen Geräten des täglichen Lebens enthalten (Broy et al., 1998). Im Bereich der Konsumgüterelektronik (Unterhaltungs- und Haushaltsgerätebereich) sind Fotoapparate, Videokameras, HiFi- und TV-Geräte, Set-Top-Boxen (Decoder für digitale TV-Programme), Waschmaschinen, Wäschetrockner, Mikrowellengeräte, Staubsauger sowie Heizungssteuerungen prominente Beispiele für Produkte, die im Allgemeinen mehrere eingebettete Systeme enthalten.

*Weitere
Beispiele*

Aus der großen Anzahl der hier aufgelisteten Produktkategorien und deren breiter Streuung über das gesamte Produktspektrum lässt sich bereits die Bedeutung heute im Einsatz befindlicher eingebetteter Systeme erahnen. Hinzu kommt, dass letztere insbesondere in Massenmärkten wie Konsumelektronik, Telekommunikation und Automobiltechnik weite Verbreitung gefunden haben und in immer größerem Umfang eingesetzt werden. Beispielsweise hat sich im Zeitraum zwischen 1985 und 1994 der Programmumfang, also der Software-Anteil, in Automobilsystemen der Firma Siemens Automotive S.A. (Toulouse) alle zwei bis drei Jahre verdoppelt, siehe (Siemens, 1994).

1.4 Begriffsdefinitionen

Zum besseren Verständnis der in diesem Buch verwendeten Begriffe geben wir im Folgenden eine Reihe entsprechender Definitionen an. Dieser Abschnitt ist insbesondere deshalb von Bedeutung, da sowohl im akademischen, als auch im industriellen Umfeld einige der Begriffe mit unterschiedlicher Bedeutung belegt sind. Die hier formulierten Definitionen beziehen sich überwiegend auf die informationsverarbeitenden Systeme (Hardware beziehungsweise Software) und weniger auf die technisch-physikalische Umgebung (Broy et al., 1998):

*Definition
(System)*

Definition (System):

Unter einem System versteht man ein mathematisches Modell S , das einem Eingangssignal der Größe x ein Ausgangssignal y der Größe $y=S(x)$ zuordnet.

Wenn das Ausgangssignal nur vom aktuellen Wert des Eingangssignals abhängt, so spricht man von einem *gedächtnislosen* System. Wenn aber dieser von vorhergehenden Eingangssignalen abhängt, so spricht man von einem *dynamischen* System.

*Definition (reak-
tives System)*

Definition (reaktives System):

Ein reaktives System kann aus Software und/oder Hardware bestehen und setzt Eingabeereignisse (deren zeitliches Auftreten meist nicht vorhergesagt werden kann) – oftmals aber nicht notwendigerweise unter Einhaltung von Zeitvorgaben – in Ausgabeereignisse um.

Definition (hybrides System):

Systeme, die sowohl kontinuierliche (analoge), als auch diskrete Datenanteile (wertkontinuierlich) verarbeiten und/oder sowohl über kontinuierliche Zeiträume (zeitkontinuierlich), als auch zu diskreten Zeitpunkten mit ihrer Umgebung interagieren, heißen hybride Systeme.

Definition (hybrides System)

Zur expliziten Unterscheidung zwischen beiden Formen hybrider Systemanteile werden die Begriffe “datenkontinuierlich/-diskret” oder „wertkontinuierlich/-diskret“ und “zeitkontinuierlich/-diskret” verwendet.

Definition (verteilt System):

Ein verteiltes System besteht aus Komponenten, die räumlich oder logisch verteilt sind und mittels einer Koppelung bzw. Vernetzung zum Erreichen der Funktionalität des Gesamtsystems beitragen.

Definition (verteilt System)

Die Entwicklung von Steuergeräten hat in der zweiten Hälfte des letzten Jahrhunderts einen bedeutenden Wechsel vollzogen. Anfängen von elektromechanischen Steuergeräten über elektrische Steuergeräte ist man heutzutage bei elektronischen (vollprogrammierbaren) Steuergeräten angelangt.

Definition (Steuergerät):

Ein Steuergerät (engl. Electronic Control Unit, kurz ECU) ist die physikalische Umsetzung eines eingebetteten Systems. Es stellt damit die Kontrolleinheit eines mechatronischen Systems dar. In mechatronischen Systemen bilden Steuergerät und Sensorik/Aktorik oft eine Einheit.

Definition (Steuergerät)

Steuergeräte sind im Prinzip wie folgt aufgebaut: Die Kernkomponente des Steuergeräts stellt ein Mikrocontroller oder Mikroprozessor (Beispiele: Power PC, Alpha PC) dar. Zusätzlich kann es optional ein externes RAM und/oder ROM besitzen sowie sonstige Peripherie und Bauelemente.

Definition (mechatronisches System, Mechatronik):

Wird Elektronik zur Steuerung und Regelung mechanischer Vorgänge räumlich eng mit den mechanischen Systembestandteilen verbunden, so sprechen wir von einem mechatronischen System. Der Forschungszweig, der sich mit den Grundlagen und der Entwicklung mechatronischer Systeme befasst, heißt Mechatronik.

Definition (mechatronisches System, Mechatronik)

Der Begriff *Mechatronik* (engl. mechatronics) ist ein Kunstwort bestehend aus *Mechanik* und *Elektronik*. Bei der Mechatronik handelt sich um ein interdisziplinäres Gebiet der Ingenieurwissenschaften, das auf Maschinenbau, Elektrotechnik und der Informatik aufbaut. Mechatronische Systeme enthalten in zunehmendem Maße hierarchisch angeordnete, untereinander gekoppelte eingebettete Systeme. Ein typisches mechatronisches System nimmt Signale auf, verarbeitet sie und gibt wiederum Signale aus, die dann ggf. in Kraft oder Bewegung umgesetzt werden.

1.5 Logischer Aufbau eingebetteter Systeme

In diesem Abschnitt charakterisieren wir die grundlegenden Bestandteile eingebetteter Systeme und erklären die daraus resultierende logische Strukturierung der Anforderungen. Nach der Lektüre dieses Abschnittes sollte der Leser den prinzipiellen logischen Aufbau eines eingebetteten Systems und dessen wesentliche Bestandteile kennen.

Beim Einsatz von Hardware in eingebetteten Systemen ist auf die *Umgebungsbedingungen am Einsatzort* besonders zu achten. Sie muss unter anderem robust gegen folgende Störfaktoren sein: Wärme bzw. Kälte, Staub, Feuchtigkeit, Spritzwasser, mechanische Schwingungen bzw. Stöße, Fremdkörper und elektromagnetische Störungen (EMS). Insbesondere in der Automobilindustrie gibt es hier genaue, oftmals Hersteller-spezifische Vorschriften, die diesbezüglich genaue Vorgaben enthalten. Sie können durch physikalische (z. B. durch Schirmung bzw. Gehäuse) und geometrische (beispielsweise Vermeidung von EMS durch Verdrillen der Kabel) Maßnahmen eingehalten werden.

Die 5 Strukturbestandteile eingebetteter Systeme

Ein eingebettetes System kann in der Regel in die folgenden fünf strukturellen Bestandteile zerlegt werden (Broy und Scholz, 1998):

- Die Kontrolleinheit bzw. das Steuergerät, d. h. das eingebettete Hardware/Software System,
- die Regelstrecke mit Aktoren (auch: Aktuatoren) und Sensoren, d. h. das gesteuerte physikalische System,
- die Benutzerschnittstelle,
- die Umgebung sowie
- den Benutzer.

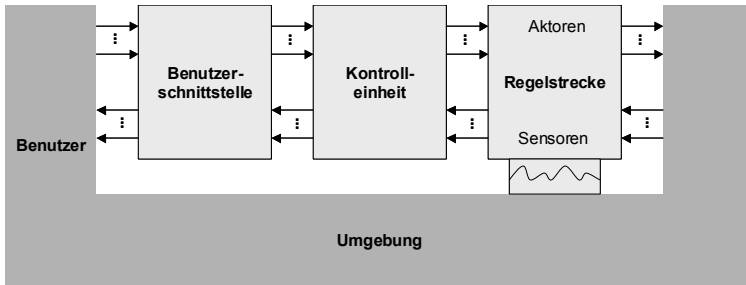


Abb. 1.2:
Logische
Referenz-
architektur eines
eingebetteten
Systems

Die *Grobarchitektur* eines eingebetteten Systems unterteilt das System in diese Bestandteile und beschreibt, wie sie miteinander verbunden sind. Abbildung 1.2 zeigt diese Architektur als Datenflussdiagramm. Die gerichteten Pfeile stellen gerichtete Kommunikationskanäle dar. Auf diesen Kanälen werden Ströme diskreter Nachrichten oder kontinuierlicher Signale („diskret“ bedeutet hier immer ereignisdiskret, „kontinuierlich“ meint zeit- und wertkontinuierlich) übermittelt. Die gezackte Linie, welche die Kommunikationskanäle zwischen Regelstrecke und Umgebung markiert, deutet an, dass diese Bestandteile auf sehr komplexe, schwer formalisierbare Weise in Wechselwirkung stehen können. Die Kontrolleinheit ist nicht direkt mit der Umgebung verbunden, sondern nur mit der Benutzerschnittstelle und der Regelstrecke. Die Grauschattierung in der Zeichnung weist darauf hin, dass wir zwischen Benutzer und Systemumgebung nicht klar trennen.

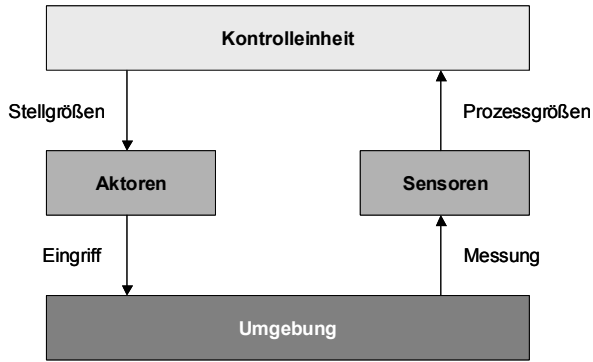
Grobarchitektur

Wie wir der Abbildung 1.2 entnehmen können, handelt es sich beim Ablauf eines eingebetteten Systems um eine geschlossene Wirkungskette, vgl. Abbildung 1.3. Die Aufgaben dieser Wirkungskette sind:

Wirkungskette

- Die Erfassung von Eingabeereignissen durch Sensoren,
- die steuernde Einwirkung auf den zu kontrollierenden Prozess durch Aktoren und
- die Umwandlung (kurz: Wandlung) analoger elektrischer Größen der Eingabeereignisse in digitale Signale zur Verarbeitung durch Rechner (die sogenannte Analog-Digital/Digital-Analog-Wandlung, kurz AD/DA-Wandlung).

Abb. 1.3:
Wirkungskette
System/
Umgebung



Der zu steuernde technische Prozess ist also über Sensoren und Aktoren an das Steuergerät gekoppelt und kommuniziert über diese mit ihm. Sensoren und Aktoren fasst man unter dem Begriff *Peripherie* oder *I/O-System* zusammen. I/O (engl. Input/Output) steht hier für Eingabe/Ausgabe (kurz: E/A).

1.5.1 Kontrolleinheit

Die Kontrolleinheit empfängt Signale von der Benutzerschnittstelle und den Sensoren an der Regelstrecke. Diese Eingaben werden verarbeitet und Antwortsignale werden an die Benutzerschnittstelle und die Aktoren an der Regelstrecke geschickt. Die Kontrolleinheit bildet den Kern des eingebetteten Systems. Selbstverständlich muss die Kontrolleinheit keine monolithische Komponente sein. Sie kann im Entwurf durchaus in ein Netzwerk von parallelen, örtlich verteilten Subkomponenten zerlegt werden, die mit Benutzerschnittstelle und Regelstrecke interagieren. Fragen der Parallelität und der Verteilung sind beim Entwurf von eingebetteten Systemen allerdings Teil der sogenannten „Glass-Box-Sicht“ auf die Kontrolleinheit. In den frühen Phasen der Systementwicklung sind wir dagegen hauptsächlich an der „Black-Box-Sicht“ interessiert. Eine direkte Verbindung von Benutzerschnittstelle und Regelstrecke kann als Spezialfall der Kontrolleinheit modelliert werden. Wie wir bereits gesehen haben, stellen Steuergeräte die Kontrolleinheit eines mechatronischen Systems dar.

Moderne PKWs der Oberklasse enthalten inzwischen bis zu 100 Steuergeräten (ECU, Electronic Control Unit), Tendenz steigend. Damit einhergehend werden ca. 3 km Kabel und ca. 2.000 Steckverbindungen verbaut. In den Steuergeräten reguliert Software

mit mehr als 600.000 LOCs (Lines of Code, Quellcodezeilen) zahlreiche Funktionen sowie deren Zusammenspiel. Das umfasst u. a. die Regelung der Bremskraft, die Zentralverriegelung des Fahrzeugs und die Funktion des Scheibenwischers (Quelle: DaimlerChrysler).

Damit ändert sich auch signifikant die Wertschöpfung im Automobilbau. 90% der Innovationen im Fahrzeug werden durch Elektronik geprägt, davon 80% im Bereich der Software. Dabei wächst der Anteil der Software an der Produkt-Wertschöpfung stetig (Quelle: Audi).

Bis 2010 prognostizieren Analysten eine jährliche Wachstumsrate für elektronische Funktionen im Auto von mehr als 10% pro Jahr. Der Elektronikanteil der Automobilproduktionskosten wird sich von 22 bis 25% im Jahre 2000 auf 35 bis 40% im Jahre 2010 erhöhen. Der Gesamtmarkt für Elektronik und Software soll in diesem Bereich um 115% bis auf 270 Mio. Euro zunehmen (Quelle: TTTech).

Grundlagen und Aufbau von Mikrocomputern bzw. Mikroprozessoren und somit der Kontrolleinheit sind nicht Inhalt dieses Buches und können beispielsweise in (Flynn et al, 2001), (Kelch, 2003), (Kelch, 2003a), (Märting, 2003), (Stallings, 2000), (Tanenbaum, 2001) oder (Zargham, 1996) studiert werden. Mikrocontroller kennzeichnen eine Klasse von Mikroprozessoren, die auf den speziellen Anwendungsbereich der Steuerung von Prozessen zugeschnitten sind. Wir betrachten im Folgenden einige Spezialfälle genauer.

Definition (Application Specific Instruction Set Processor, ASIP):

Ein ASIP (applikationsspezifischer Prozessor) ist ein Prozessor, der von seiner Struktur als auch von seinem Befehlssatz her auf seinen Einsatz für bestimmte Anwendungen hin optimiert ist.

*Definition
(ASIP)*

ASIPs besitzen spezielle Instruktionssätze, funktionale Einheiten, Register und spezielle Verbindungsstrukturen. Diese Spezialisierung führt zu Architekturen, die sich von denen eines Mikroprozessors stark unterscheiden. ASIPs stellen hinsichtlich Flexibilität und Performanz die Nahtstelle von der Software- zur Hardwareseite her. Aus Kostengründen ist ein ASIP oft nur ein um verschiedene Umfänge reduzierter Prozessor und damit günstiger als ein „Vielzweckprozessor“ (engl. multi purpose processor), aber aufgrund seiner Programmierbarkeit immer noch flexibler als dedizierte Hardware. Er leistet eine höhere Verarbeitungsgeschwindigkeit und benötigt eine geringere Leistungsaufnahme aufgrund

optimierter Strukturen. Sein Nachteil ist die komplizierte und aufwändige Entwicklung.

Zu der Klasse anwendungsspezifischer Prozessoren gehört auch die Klasse der digitalen Signalprozessoren (DSPs).

*Definition
(DSP)*

Definition (Digitaler Signal Prozessor, DSP):

Ein DSP ist ein spezieller Mikroprozessor, der Befehle (und damit Verarbeitungseinheiten) zur Durchführung von Signalverarbeitungsaufgaben besitzt. Er ist dabei auf die häufig vorkommenden Operationen bei der digitalen Signalverarbeitung wie z. B. die schnelle Fourier Transformation (FFT) spezialisiert. Der Effizienz der Operationen kommt hierbei ein hoher Stellenwert zu.

Digitale Signal Prozessoren werden beispielsweise in MP3-Decodern, bei der Bildverarbeitung sowie in der Sprachsignalverarbeitung eingesetzt.

*Definition
(ASIC)*

Definition (Application Specific Integrated Circuit, ASIC):

Ein ASIC ist eine anwendungsspezifische integrierte Schaltung (engl. Integrated Circuit, kurz: IC), die für spezielle Anwendungen entwickelt wurde und spezielle Schaltungen enthält, die hohe Leistungs- und Zeitanforderungen erfüllen können. Er kann wie ein ASIP programmierbar oder auch festverdrahtet sein.

*Definition
(FPGA)*

Definition (Field Programmable Gate Array, FPGA):

Das Field Programmable Gate Array (FPGA) ist ein komplexer, programmierbarer Logikbaustein, der zum Aufbau digitaler, logischer Schaltungen dient. Er besteht im Wesentlichen aus einzelnen Funktionsblöcken, die in einer regelmäßigen Struktur (engl. Array) angeordnet sind und einem Netzwerk von Verbindungen zwischen diesen Blöcken. Die speziellen Funktionen der einzelnen Blöcke und die Auswahl der benötigten Verbindungen sind programmierbar.

Bei FPGAs wird die Implementierung von logischen Funktionen hauptsächlich durch die Programmierung der Verbindungsleitungen zwischen den Logikblöcken erreicht. Damit kann in das FPGA eine bestimmte vom Anwender entwickelte Schaltung implementiert werden. Man spricht hier auch von einer Personalisierung des Bausteins.

FPGA-Arten

Es gibt zwei verschiedene Arten von FPGAs, rekonfigurierbare sowie nicht-rekonfigurierbare. Für die erstgenannten verwendet man flüchtige Speichertechnologien wie SRAM. Ihr Nachteil ist, dass die Programmierung nur solange im Baustein anhält, wie auch eine

Stromversorgung besteht. Wird sie unterbrochen, so muss der Baustein neu konfiguriert (rekonfiguriert) werden bzw. seine Konfiguration neu eingelesen werden.

Die nicht-rekonfigurierbaren FPGAs können dagegen nur einmal programmiert werden, behalten ihre Programmierung dafür aber für immer und unabhängig von der Stromversorgung. Dies gelingt meist durch die physikalische Zerstörung nicht gebrauchter Leitungen; die Schaltungskonfiguration ergibt sich somit indirekt.

Im Gegensatz zu gewöhnlichen Gate Arrays (GA) sind FPGAs *programmierbare* Logikbausteine, deren Funktionalität durch das Zusammenschalten verschiedener Funktionsblöcke erreicht wird. Während PLAs (engl. Programmable Logic Arrays) nur Boolesche Funktionen in zweistufiger Form realisieren können, kann man mit FPGAs auch Speicherzellen realisieren und so eignen sich FPGAs auch zur Realisierung von Steuerwerken (in Form endlicher Automaten).

1.5.2 Regelstrecke

Die Regelstrecke zeichnet sich im Allgemeinen durch ihren heterogenen Charakter aus. Sie kann aus elektronischen, elektrischen und mechanischen Teilen bestehen. Anders als in der Regelungstechnik betrachten wir beim Softwareentwurf von eingebetteten Systemen die Sensoren und Aktoren als Teil der Regelstrecke. Zumindest aus der Sicht der Kontrolleinheit kann die Regelstrecke daher als Komponente gesehen werden, die diskrete und/oder kontinuierliche Kontrollsignale erhält und über ihre Sensoren diskrete und/oder kontinuierliche Signale ausgibt.

Zusätzlich steht die Regelstrecke auch mit ihrer Umgebung in Wechselwirkung. Es ist eine wesentliche und manchmal schwierige Entscheidung des Systemanalytikers, die Grenze zwischen Regelstrecke und Umgebung festzulegen. Wenn diese Grenze bestimmt ist, muss die Interaktion zwischen Umgebung und Regelstrecke modelliert werden.

Diese Wechselwirkung kann durch komplexe kausale Abhängigkeiten zwischen komplizierten physikalischen oder chemischen Prozessen charakterisiert sein, wie z. B. beim Motormanagement von Kraftfahrzeugen. Daher kann es in manchen Fällen sehr schwierig sein, ein Modell zu finden, in dem die Kommunikation zwischen Regelstrecke und Umgebung dargestellt werden kann.

Wir nehmen deshalb vereinfachend an, dass die Interaktion zwischen beiden Komponenten allein durch eine Zahl diskreter

und/oder kontinuierlicher Ströme modelliert werden kann. Diskrete Ströme von Nachrichten codieren dabei Ereignisse, kontinuierliche Ströme (Funktionen über der Zeit) modellieren sich stetig ändernde Parameter des Systemzustandes. Unterschiedliche mögliche Reaktionen (man spricht hier auch von „Unterspezifikation“) der Umgebung oder Unwissen über ihre genauen Reaktionen können durch Nichtdeterminismus oder durch Prädikate, die Mengen von möglichen Verhalten beschreiben, modelliert werden.

In Fällen, in denen ein vereinfachtes Modell nach diesem Muster nicht geeignet ist, kann versucht werden, die Regelstrecke sehr klein zu wählen, oder allein die Interaktion von Kontrolleinheit und Regelstrecke ohne die Umgebung zu untersuchen. Der Einfluss von externen Störungen muss dann gesondert betrachtet werden. Eine andere Möglichkeit ist, die Regelstrecke wesentlich größer zu wählen und einen großen Teil der Umgebung in sie zu integrieren. Dieser Ansatz kann dann zu einer leichter handhabbaren Schnittstellenbeschreibung der Regelstrecke führen. Schließlich kann es in manchen Fällen auch notwendig sein, mit Wahrscheinlichkeiten zu arbeiten, und die Systemanforderungen mit Wahrscheinlichkeiten auszudrücken.

1.5.2.1 **Peripherie**

Sensoren und Aktoren können sowohl jeweils *einzel*n als auch über *Bussysteme* verschiedenartiger Topologie (Bus, Ring usw.) mit Steuergeräten verbunden sein.

Die Kommunikationsschnittstelle zwischen Steuergerät und technischem Prozess ist in Abbildung 1.3 stark vereinfacht dargestellt. Sie zeigt den Spezialfall der *digitalen Schnittstelle*, ist aber Grundlage sowohl für den Aufbau als auch für das Verständnis für alle weiteren Ein-/Ausgabeschnittstellen. Bei ihr handelt es sich um die einfachste Ein-/Ausgabemöglichkeit, weil das Steuergerät selbst mithilfe eines speziellen I/O-Registers (Eingabe-/Ausgabe-Registers) über digitale elektrische Signale mit dem Prozess kommuniziert.

Gängige Spannungen für die *digitale Ein- bzw. Ausgabe* sind beispielsweise 0Volt/230Volt, 0V/5V bei TTL-Technologie oder 0V/12V im Falle des Bordspannungsnetzes eines Kraftfahrzeugs.

Da die im realen technischen Prozess vorkommenden Signale rein analoger Form sind, Steuergeräte genauso wie andere Mikrocomputer aber nur digitale Signale verarbeiten können, sind weitere technische Hilfsmittel notwendig, um eine *Konvertierung*

(Wandlung) der Signale (von analog nach digital und vice versa) vornehmen zu können.

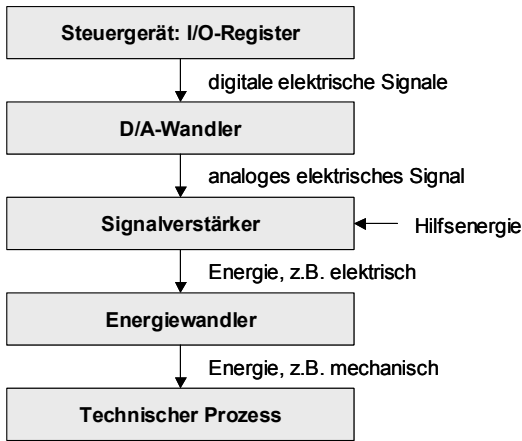
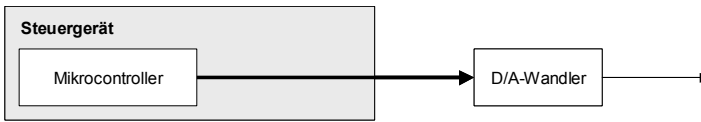


Abb. 1.4:
Signal-
verarbeitungs-
kette mit
D/A-Wandler

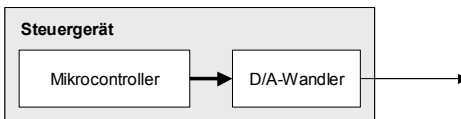
1.5.2.2 Digital/Analog-Wandler

Digital/Analog-Wandler (kurz: D/A-Wandler) erzeugen aus digitalen Signalen ein analoges Signal (Spannung). Mit ihrer Hilfe lässt sich eine komplexere Signalverarbeitungskette darstellen (vgl. Abbildung 1.4).

D/A-Wandler können auf verschiedene Weise in die Wirkungskette zwischen Steuergerät und Sensor bzw. Aktor integriert sein; Abbildung 1.5 beschreibt diese.



(a) Externer D/A-Wandler



(b) D/A-Wandler und Mikrocontroller auf Baugruppe



(c) D/A-Wandler als Teil des Mikrocontrollers

Abb. 1.5:
Integration von
D/A-Wandlern,
nach
(Bender, 2003)

Im ersten Fall (a) handelt es sich beim D/A-Wandler um eine eigene Komponente, die außerhalb des Steuergeräts angesiedelt ist. Im zweiten Fall (b) ist der D/A-Wandler zusammen mit dem Mikrokontroller auf derselben Baugruppe innerhalb des Steuergeräts aufgebracht. Die höchste Form der Integration wird erreicht, wenn (c) der D/A-Wandler Teil des Mikrocontrollers selbst ist.

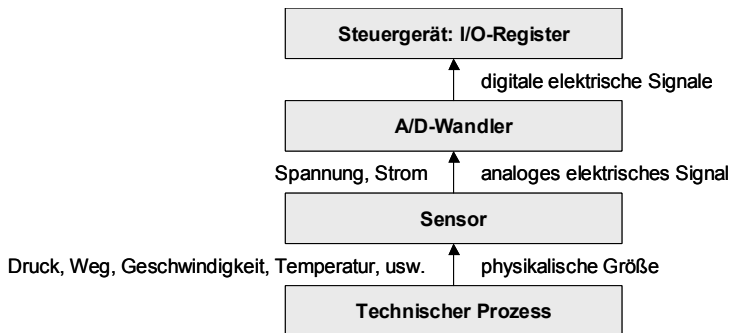
*PCM-Wandler,
DM-Wandler*

Eine weitere Möglichkeit, Wandler zu unterscheiden, ist in PCM- und DM-Wandler. *PCM-Wandler* (PCM = Pulse Code Modulation) sind Wandler, bei denen am Ausgang eine duale Codierung des Analogwertes am Eingang, also der Eingangsspannung anliegt. Die Ausgabe eines *DM-Wandlers* (Delta Modulation) beschreibt die Differenz zum Wert des vorangegangenen Abtastzeitpunktes. Das Wandler-Ausgangssignal codiert also eine Differenz, die aus den Werten des Eingangssignals zwischen zwei Abtastzeitpunkten bestimmt wird. Der Vorteil der Delta Modulation liegt in der Datenreduktion, die besonders wichtig bei höheren Abtastraten wie z. B. von Audio-, oder Videosignalen wird. Die grundlegende Idee hierbei ist, dass die zu wandelnden Signale i.d.R. eine hohe Datenredundanz aufweisen, weil benachbarte Abtastwerte nur geringfügig voneinander abweichen.

1.5.2.3 **Analog-/Digital-Wandler**

Analog/Digital-Wandler (kurz: A/D-Wandler) erzeugen aus einem analogen Signal (in Form von Spannung oder Strom) digitale Signale. Ihre Funktionsweise ist analog zu D/A-Wandlern. Auch sie dienen zur Abbildung komplexerer Signalverarbeitungsketten (siehe Abbildung 1.6). Hinsichtlich ihrer Integration in diese Wirkungskette sind die gleichen drei Alternativen wie beim D/A-Wandler möglich (vgl. Abbildung 1.5).

*Abb. 1.6:
Signalverarbeitungskette mit
A/D-Wandler*



1.5.2.4 Sensoren

Grundsätzlich unterscheidet man zweierlei Betriebsarten, wie das Steuergerät Informationen in Form von Signalen von Sensoren erhält: Den Interrupt-Betrieb sowie den Polling-Betrieb.

Beim *Interrupt-Betrieb* führt ein vom Sensor aufgenommenes Ereignis dazu, dass das Steuergerät seine aktuelle Verarbeitung unterbricht (engl. to interrupt) und auf das Ereignis reagiert. Das Steuergerät nimmt hier eine passive Rolle ein.

Beim *Polling-Betrieb* fragt das Steuergerät in regelmäßigen, meist sehr kurz aufeinander folgenden zeitlichen Abständen selbst aktiv beim Sensor nach, ob ein neues Ereignis anliegt (engl. request) und erhält eine Antwort (engl. response) vom Sensor.

Definition (Sensor), nach (Schürmann, 2001):

Ein Sensor ist eine Einrichtung zum Feststellen von physikalischen oder chemischen Eingangsgrößen, die optional eine Messwertzuordnung (Skalierung) der Größen treffen kann, sowie ggf. ein digitales bzw. digitalisierbares Ausgangssignal liefern kann.

*Definition
(Sensor)*

Der Begriff Sensor kommt dabei vom lateinischen „Sensus“, der Sinn. Die IEC (International Electrotechnical Commission) definiert den Begriff Sensor wie folgt: Das primäre Element in einer Messkette, das eine variable, im Allgemeinen nichtelektrische Eingangsgröße in ein geeignetes Messsignal, insbesondere einer elektrischen, umsetzt.

Der Sensor ist damit der Spezialfall des „Transducers“, der eine Energieform in eine andere umwandelt. Ein Sensor ist im Allgemeinen ein preiswerter und zuverlässiger Messwertaufnehmer mit ausreichender Genauigkeit, der für die Massenherstellung geeignet ist.

Definition (Sensorik)

Unter Sensorik versteht man die Signalaufnahme von der Umwelt mit optionaler Signalverarbeitung für den Einzelsensor oder Sensorensysteme.

*Definition
(Sensorik)*

Vorsicht: Unter Sensorik wird in anderen naturwissenschaftlichen Teilgebieten oft auch der Einsatz menschlicher Sinne (Riechen, Schmecken usw.) zu Prüf- und Messzwecken verstanden.

Sensoren können nach unterschiedlichen Maßgaben klassifiziert werden. Eine erste Möglichkeit ist nach der *Ausgabe*, der *Messgröße* und dem *Wirkungsprinzip* (Schürmann, 2001).

*Klassifizierung
von Sensoren
nach Ausgabe,
Messgröße,
Wirkungsprinzip*

Betrachten wir zunächst die *Art der Ausgabe* am Ausgang des Sensors. Ein Sensor kann mechanische, pneumatische, hydraulische oder elektrische Werte liefern, wobei wir im Folgenden ausschließlich elektrische Werte betrachten werden.

Bei der Art der *Messgröße* können wir nach Weg, Winkel, Kraft, Druck, Beschleunigung, Temperatur, Gas, usw. unterscheiden.

Mögliche *Wirkungsprinzipien* des Sensors sind ohmsch, kapazitiv, induktiv, optoelektronisch, thermoelektrisch, piezoelektrisch, etc.

Klassifikation von Sensoren nach Energienutzung

Das gemeinsame Merkmal aller Sensortypen ist die Verknüpfung der Messgröße mit einer elektrischen Ausgangsgröße. Ihr Unterscheidungsmerkmal ist die *Art der Nutzung der Energie* der Messgröße, die in elektrische Energie gewandelt wird:

- **Rezeptiver Sensor:** Nimmt ein vorhandenes Signal auf und wandelt ggf. nur um. Die Signalverarbeitung erfolgt anschließend. Rezeptive Sensoren werden auch passive Sensoren bzw. Wandler genannt. Beispiele für sie sind Kameras und Mikrophone.
- **Signalbearbeitender Sensor:** Stimuliert die Umwelt und nimmt die „Antwort“ auf. Er benötigt hierfür eine zusätzliche Energie- oder Signalquelle. Beispiele sind Ultraschall-Sensoren oder Laser-Scanner. Signalbearbeitende Sensoren werden auch als aktive Sensoren bzw. Wandler bezeichnet.

Klassifikation von Sensoren nach Anwendung

Eine weitere Klassifikationsmöglichkeit für Sensoren ergibt sich nach der *Art ihrer Anwendung* (Schürmann, 2001):

- **Interne Sensoren** sind Sensoren, welche die inneren Zustände (Position, Geschwindigkeit, Beschleunigung, Orientierung) einer Maschine erfassen. Beispiele hierfür sind das Erfassen der Gelenkstellung, der Gelenkgeschwindigkeit und der Gelenkbeschleunigung eines Roboterarms.
- **Externe Sensoren** dagegen erfassen (aus der Sicht des Gesamtsystems) äußere Zustände, also Zustände der Umwelt. Beispiele hierfür sind das Messen des Abstands zwischen Fahrzeug und umgebenden Hindernissen (beispielsweise andere Fahrzeuge) z. B. mit Radar, die (optische) Werkstückidentifikation an einem Fertigungsabschnitt oder das Feststellen des Tankfüllstand eines Kraftfahrzeugs.



Die letzte Klassifikationsmöglichkeit ergibt sich durch die *Analyse des Wertebereichs* der Sensoren (Schürmann, 2001): Hier kann zwischen binären (Beispiel: Lichtschranke), skalaren (Beispiele: Winkel-, Abstandssensor), vektoriellen (Beispiel: Gyrosensor) und Muster-erkennenden Sensoren (Beispiele: Laserscanner, Videocamera) unterschieden werden.

Klassifikation von Sensoren nach Wertebereich

1.5.2.5 Aktuatoren

Aktuatoren bzw. Aktoren sind die Verbindungsglieder zwischen dem informationsverarbeitenden Teil eines eingebetteten Systems und dem Prozess. Sie wandeln die Energie, die an ihrem Eingang anliegt, in mechanische Arbeit um.

Beispiele für Aktuatoren sind Elektromotoren, Elektromagneten, hydraulische Aktuatoren, piezoelektrische Aktuatoren u.v.m.

Üblicherweise sind Aktuatoren mit *Standardschnittstellen* (wie etwa RS232 oder USB) ausgestattet, über die, mehr oder weniger aufwändig, eine Systemsteuerung möglich ist. Der Umfang der möglichen Instruktionen ist meist Sensor-abhängig, zur Erstellung eines Steuerungsablaufs ist ein genaues Studium der Sensorbeschreibung zwingend. In den ersten Generationen intelligenter Aktoren wurden aus Performanzgründen oft für den menschlichen Benutzer kryptische Kommandos aus Binärsequenzen verwendet. Heutzutage sind dies textuelle Kommandos, die weitgehend selbsterklärend sind.

1.5.3 Benutzerschnittstelle

Die meisten eingebetteten Systeme interagieren nicht nur mit der Regelstrecke, sondern reagieren auch auf Benutzereingaben und liefern Ausgaben, um den Benutzer über den gegenwärtigen Systemzustand zu informieren. Diese Kommunikationsaufgaben übernimmt die Benutzerschnittstelle.

Ebenso wie die Regelstrecke kann auch die Benutzerschnittstelle nicht als ein reines Hardware oder Software System gesehen werden. Im Allgemeinen besteht es aus einer Menge von Eingabegeräten (Schaltern, Tasten, Joysticks, etc.) und Ausgabegeräten (Signallichter, Bildschirmen, etc.). Oft gibt es Teile des Gesamtsystems, die sowohl als Teil der Benutzerschnittstelle, als auch als Teil der Regelstrecke gesehen werden können. Auch hier muss der Systemanalytiker eine geeignete Aufteilung festlegen. Natürlich kann auch

die Benutzerschnittstelle in eine Menge von dezentralen Bedienstationen zerlegt werden.

1.6 Softwareentwicklung eingebetteter Systeme

Die Entwicklung der Software eingebetteter Systeme ist eine komplexe Aufgabe, die derzeit noch nicht in allen Teilen zufriedenstellend beherrscht wird. Um die korrekte Funktionsweise solcher Systeme und kurze Entwicklungszeiten zu gewährleisten, ist es erforderlich, den Systementwickler in allen Phasen der Entwicklung möglichst gut methodisch und technisch durch ein passendes Werkzeug (CASE-Tool, CASE = Computer Aided Software Engineering) zu unterstützen.

1.6.1 Motivation

Die Software in eingebetteten Systemen muss sorgfältig auf Verbraucherwünsche und -anforderungen abgestimmt werden. Beta-Tests, wie sie etwa in der PC-Standardsoftwareindustrie weit verbreitet sind, können zum Test von Software in eingebetteten Systemen nicht verwendet werden. Kunden können im täglichen Verkehr schließlich kaum als Testpersonen fungieren. Aus diesen Gründen erfordert die Entwicklung von Software für eingebettete Systeme große Umsicht und Sorgfalt. Mittel- und langfristig wird es nur dann gelingen, zuverlässige reaktive, eingebettete Systeme herzustellen, wenn dem Entwickler ein speziell darauf angepasster Entwicklungsprozess zur Verfügung steht. Aufgrund der hohen Anforderungen an Nutzernähe und Zuverlässigkeit ist es unerlässlich, dass alle Konzepte und Anforderungen eines reaktiven Systems so früh wie möglich und beständig im Entwicklungsprozess validiert und verifiziert werden.

1.6.2

Begriffsklärung

Unter einer *Spezifikation* versteht man die Festlegung (formal mit eindeutiger Semantik) was das System können muss. Dies umfasst das gewünschte Verhalten ebenso wie die Schnittstellen (Zahl und Art der Ein-/Ausgänge, evtl. sogar Semantik) sowie evtl. auch Vorgaben bzgl. Geschwindigkeit, Kosten, Fläche, Leistungsverbrauch usw.

Spezifikation

Die *Korrektheit* (oder der Nutzen) eines eingebetteten Systems muss durch die Beobachtung des Verhaltens des Gesamtsystems, also der Reaktionen auf Eingaben von Benutzer und Umgebung, bestimmt werden. Wir sind normalerweise nicht am Verhalten der Kontrolleinheit, sondern vielmehr am Verhalten der Regelstrecke unter dem Einfluss der Kontrolleinheit interessiert. Wenn die Regelstrecke unerwünschtes Verhalten zeigt, so ist der Entwurf ungeeignet. Beobachtungen müssen dabei auch die Eingaben von Benutzer und Umgebung umfassen. Nur relativ zu diesen Eingaben ist es möglich, festzulegen welches Verhalten die Regelstrecke jeweils zeigen soll. Mögliche Gründe für falsches oder unerwartetes Verhalten sind: Die Annahmen über das Verhalten der Regelstrecke sind falsch; die Kontrolleinheit sendet nicht die richtigen Signale oder sendet sie nicht rechtzeitig; die Benutzerschnittstelle übermittelt nicht die richtige Information zwischen Benutzer und Kontrolleinheit.

Korrektheit

Mit dem Begriff der *Implementierung* meinen wir die reine Umsetzung der Spezifikation in eine konkrete Realisierung des zu entwerfenden Systems.

Implementierung

1.6.3

Entwurf

Der Entwurf eingebetteter Systeme ist durch seine hohe Komplexität gekennzeichnet. Diese Komplexität kann durch einen hierarchischen Entwurfsstil mit Simulation und Optimierung auf mehreren Ebenen bewältigt werden. Den einzelnen Ebenen werden spezifische Modellvorstellungen (Sichten, Eigenschaften, Domänen) zugeordnet.

Software wird nicht in einem einzigen Schritt entwickelt, sondern man unterteilt den Entwicklungsprozess für Software in mehrere Phasen. Hierbei haben sich einige Standardvorgehensweisen wie beispielsweise das Wasserfall-, Spiral-, oder V-Modell etabliert. Man unterteilt solche Prozesse beispielsweise in die Phasen

Anforderungserfassung (Requirements Capture), Systemanalyse, Entwurf (Design), Validierung, Verifikation, Prototypengenerierung, Simulation und Implementierung.

Bei der Entwicklung eingebetteter Systeme ist die Spezifikation der geforderten Funktionalität eine besonders kritische Aufgabe. Statistiken zeigen, dass in typischen Anwendungsgebieten mehr als 50% der Fehler, die in den ausgelieferten Systemen auftauchen und von den Kunden gemeldet werden, keine Fehler bei der Implementierung, sondern konzeptionelle Fehler in den erarbeiteten Anforderungen sind. Eingebettete Systeme, insbesondere solche in sicherheitskritischen Bereichen, erfordern ein hohes Maß an Zuverlässigkeit. Basis dieser Zuverlässigkeit ist die sorgfältige Erfassung der Anforderungen der Benutzer. In diesem Beitrag konzentrieren wir uns daher auf die beiden Phasen Anforderungsanalyse und Entwurf. In diesen Phasen spielen Modellierungstechniken eine wichtige Rolle. Folgende Aspekte stellen bei solchen Techniken eine besondere Herausforderung dar:

- Modellierung von Verhalten (digital, analog) und Zeit (diskret, kontinuierlich);
- Modularität; Integration verschiedener Modelle;
- Abstraktion und Verfeinerung; Eignung für (werkzeugunterstützte) Validierung und Verifikation;
- Anschluss an eine automatische Codegenerierung.

1.7 Besondere Herausforderungen

Beobachtungen

Bei der näheren Betrachtung der gegenwärtigen Situation der Entwicklung eingebetteter Systeme ergeben sich folgende Beobachtungen (Broy und Scholz, 1998):

- Der Kostenanteil der Software im Vergleich zur Hardware steigt.
- Aufgabenkomplexität und Vernetzungsgrad nehmen zu.
- Die Komplexität der Steuergeräte wächst exponentiell und gleichzeitig sollen Entwurfszeiten signifikant verkürzt werden.
- Eingebettete Systeme interagieren und verschmelzen mit anderen Softwaresystemen.
- Anwendungsgebiete und Funktionsumfang erweitern sich.

- Neuartige Modellierungsparadigmen sind zur Komplexitätsbewältigung erforderlich.
- Verschiedene Disziplinen wie etwa Kontroll- und Regelungstechnik, Schaltungstechnik, Software Engineering, Betriebssysteme, Compilerverbau und Maschinenwesen wachsen zusammen.

Auf Basis dieser Beobachtungen ergeben sich für die Zukunft im Bereich des Software Engineering für eingebettete Systeme folgende Herausforderungen (Broy und Scholz, 1998):

Herausforderungen

- Die „Time to Market“ muss verkürzt und die Nutzungsadäquanz (Nutzerakzeptanz) erhöht werden.
- Die Beherrschbarkeit des Entwicklungsprozesses (Kosten, Termine, Qualität) ist zu verbessern.
- Die Durchgängigkeit des Entwicklungsprozesses muss hergestellt werden.
- Die Modellierung, insbesondere von Zeit-, Reaktivitäts- und Verteilungsaspekten muss verbessert werden.
- Die Entscheidung, welche Teile des Systementwurfs in Hardware und welche in Software realisiert werden sollen, muss so spät wie möglich im Entwicklungsprozess fallen (Stichwort „Hardware/Software-Codesign“).
- Sicherheit, Zuverlässigkeit, Vorhersagbarkeit, Adaptierbarkeit, Modifizierbarkeit (Design to Change) der Entwicklungen sind zu erhöhen und eine größere Arbeitsteilung ist zu gewährleisten.
- Modellierungskonzepte für Verteiltheit, Mobilität und Dynamik sind nötig.

1.8 Zusammenfassung

Eingebettete Systeme sind in zahlreichen Gebieten des täglichen Lebens zu finden. Sie fallen in die Klasse der reaktiven Systeme und grenzen sich daher von transformationellen und interaktiven Systemen deutlich ab. Eingebettete Systeme müssen teilweise in Echtzeit bestimmte Anforderungen lösen. In diesem Fall spricht man von Echtzeitsystemen. Aufgrund dieser Unterschiede besitzen eingebettete Systeme andere Eigenschaften als beispielsweise Büroanwendungen; dies muss bei ihrer Entwicklung unbedingt beachtet werden. Die aktuellen Herausforderungen bei der Entwicklung von

Embedded Systems liegen vor allem im Bereich der Software und sind weniger auf dem Gebiet der Hardware zu suchen.

Ein eingebettetes System kann in der Regel in folgende fünf logische Bestandteile zerlegt werden: Die Kontrolleinheit, die Regelstrecke, die Benutzerschnittstelle, die technische Systemumgebung sowie der menschliche Systembenutzer. Sie ergeben zusammen mit der Beschreibung, wie diese Bestandteile miteinander verbunden sind, die Grobarchitektur jedes eingebetteten Systems.

Zum Bau eingebetteter Systeme sind hardwareseitig Steuergeräte sowie eine auf die spezifische Steuerungsaufgabe angepasste Sensorik bzw. Aktuatorik erforderlich, zu deren Realisierung neben unterschiedlichen Sensoren bzw. Aktoren (auch: Aktuatoren) auch sogenannte Wandler nötig sind. Die Verbindung von Sensor und Aktor geschieht durch ein „intelligentes“ System zur Regelung eines Prozesses, das Steuergerät. Steuergeräte können in unterschiedlicher Ausprägung vorkommen: Durch komplexe Mikroelektronik können beispielsweise Sensor und Steuergerät als intelligentes System auf einem einzigen Chip integriert werden.