
2.3 Codierung und Modulation

2.3.1 Begriffe, Ziele

Eine Codierung kann unter anderem diese **Ziele** verfolgen:

- Darstellung alphanumerischer **Zeichen** in einer standardisierten Form. Beispielsweise wird das lateinische Alphabet häufig durch den ASCII-Code dargestellt (→ Bild 2.7).
- Codierung von **Zahlen**. Zahlenwerte können prozessorintern unterschiedlich dargestellt sein. Für ihre Übertragung in Netzen sind ebenfalls standardisierte Codes erforderlich.
- Codierung von **Symbolen** (beliebige Zeichen, einzelne Bits oder Bitketten) zur Übertragung über einen physikalischen Kanal.

- Codierung von **Signalen** der realen Welt (z. B. Sprache, Bilder) so, dass zur Übertragung eine möglichst geringe Bandbreite benötigt wird.

Originaldarstellung	Codierte Darstellung Beispiel: ASCII
A	1000001
B	1000010
C	1000011
...	...
Z	1011010

Codierung

Bild 2.7 Codierung Beispiel ASCII-Code

Reale Signale (z. B. Sprache: Schallwellen, Bilder: zweidimensionale Verteilung der Lichtintensität) werden in der Regel zuerst in eine analoge elektrische Darstellung gebracht. Die Spannungsverläufe sind also kontinuierlich. Für die Verarbeitung und Übertragung werden jedoch meistens und noch zunehmend digitale (wertediskrete) Darstellungen bevorzugt. Zur Umwandlung zwischen den Signaldarstellungen existieren verschiedene Verfahren (→ Bild 2.8).

Daten: Umwandlung: Signal: Anwendungsbeispiel:

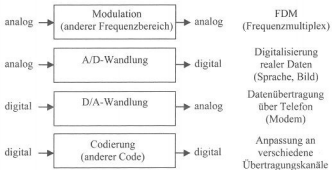


Bild 2.8 Möglichkeiten zur Signalumwandlung

Im Hinblick auf die Signalübertragung unterscheidet man die folgenden Codierungsarten: Leitungscodierung, Kanalcodierung und Quellencodierung.

Die Codierungsarten erfüllen unterschiedliche Aufgaben (→ Bild 2.9). Die gestellten Anforderungen unterscheiden sich ebenfalls. Wenn alle Codierungsarten eingesetzt werden (beispielsweise bei der Sprach- oder Bildübertragung über Paketnetzwerke) ist beim Sender die Folge Quellen-, Kanal- und Leitungscodierung zu durchlaufen (→ Bild 2.10). Beim Empfänger ist die Reihenfolge umgekehrt.

	Quellencodierung	Kanalcodierung	Leitungscodierung
Ziel	Redundanzreduktion	Erkennung und Korrektur von Fehlern	Anpassung der Codesymbole an den physikalischen Kanal
Anforderungen	<ul style="list-style-type: none"> - Verlustlose Signalreproduktion (Redundanzreduktion) bzw. - Verlustbehaftete Signalreproduktion (Irrelevanzreduktion) 	<ul style="list-style-type: none"> - Grad der Redundanz entscheidet: <ul style="list-style-type: none"> - wie viele Fehler sind erkennbar? - wie viele Fehler sind korrigierbar? - Berechnungsaufwand 	<ul style="list-style-type: none"> - Bandbreiteneffizienz - Taktregenerierbarkeit - Gleichstromfreiheit

Bild 2.9 Begriffe zur Codierung (→ Abschnitte 2.3.2 ff.)

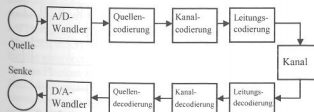


Bild 2.10 Einsatz von Codierungsverfahren in einem Kommunikationssystem

2.3.2 Leitungscodierung

Die **Leitungscodierung** (*line coding*) ordnet einem oder mehreren Bits ein bestimmtes **Symbol** (**Signalelement**) zu, das auf der Leitung übertragen wird.

Dabei sollen die folgenden Kriterien bestmöglich erfüllt werden:

- **Bandbreiteneffizienz:** Die für eine vorgegebene Datenrate (Anzahl der pro Zeiteinheit zu übertragenden Datenbits) erforderliche Bandbreite soll möglichst gering sein bzw. die auf einer Leitung mit gegebener Bandbreite mögliche Datenrate soll maximal sein.
- **Taktregenerierbarkeit:** Der Empfänger muss die Symbolzeiten (die Anfangs- und Endzeitpunkte eines Symbols) aus dem empfangenen Signal wiedergewinnen, da ihm der ursprüngliche Sendetakt nicht zur Verfügung steht.
- **Gleichspannungsanteil:** Bei der Übertragung über Leitungen, die keine Gleichspannungen übertragen können (deren untere Grenzfrequenz also größer als null ist), darf im Frequenzspektrum des Sendesignals der Wert null nicht vorkommen. Andernfalls ist es für den Empfänger schwierig bis unmöglich, das Signal korrekt zu erkennen.

Im einfachsten Fall werden für Leitungscodes zweiwertige Symbole mit unipolaren (U_+ , 0) oder bipolaren (U_+ , U_-) Spannungspegeln verwendet. Der Spannungspegel kann für die Symboldauer konstant sein oder er kann sich zu bestimmten Zeitpunkten ändern. Für verfeinerte Codes werden mehrwertige (ternäre, quaternäre) Spannungspegel verwendet. Zudem können Bitketten der Länge n in Symbolketten der Länge m ($m \neq n$) abgebildet werden. Für einige wichtige Codes sind Zeitdiagramme der Pegelverläufe in Bild 2.11 zusammengestellt.

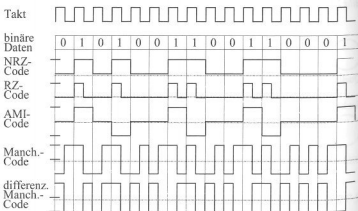
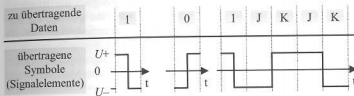


Bild 2.11 Leitungscodes

Der **NRZ-Code** (Non-Return to Zero) verwendet als Signalelement eine Spannung von (beispielsweise) 0 V für eine logische 0 und (beispielsweise) +5 V für eine logische 1. Beide Signalelemente weisen für die Symboldauer, die hier der Bitdauer entspricht, konstante Werte auf. Beim **RZ-Code** (Return to Zero) ist das Signalelement für eine logische 1 der im Bild 2.11 dargestellte Impuls (+5 V für die erste Hälfte der Symboldauer, 0 V für die zweite Hälfte). Beim **AMI-Code** (Alternate Mark Inversion) wird als Symbol für eine logische 1 abwechselnd ein konstanter Wert von +5 V bzw. -5 V verwendet.



- Signalelemente für 0, 1: Flanke zur Bitmitte ($U_+ \rightarrow U_-$ oder umgekehrt)
- Codeverletzung (code violation) J, K: keine Flanke

Bild 2.12 Begriffe Bit, Symbol, Signalelement (Beispiel: Manchester-Code)

Bild 2.12 zeigt Signalelemente des Manchester-Codes, die bipolare Spannungspegel U_+ und U_- sowie eine Flanke zur Bitmitte (deshalb auch als Biphas-Code bezeichnet) aufweisen. Die Signalelemente J und K entsprechen nicht dieser Definition und werden deshalb als **Codeverletzungen** (code violation) bezeichnet. Diese werden für besonders wichtige Kennzeichnungen eingesetzt, z. B. Rahmenanfang oder -ende (\rightarrow Token Ring, Abschnitt 6.3.3). Die Manchester-Codierung setzt für jedes zu codierende Bit das entsprechende Signalelement ein (\rightarrow Bild 2.11). Dabei können sich weitere Flanken zwischen aufeinander folgenden Signalelementen ergeben. Beim differenziellen Manchester-Code hängt die Auswahl eines Signalelements vom vorhergehenden Signalelement wie folgt ab: für eine logische 0 muss zum Symbolanfang eine Flanke (Übergang 0 zu 1 oder umgekehrt) entstehen, bei einer logischen 1 darf hingegen keine Flanke auftreten.

Der 4B5B-Code ist ein Beispiel eines (m -zu- n)-Codes. Dabei werden 16 verschiedene Bitkombinationen auf 32 Codewörter abgebildet. Die Hälfte der 32 Codewörter kann demnach für die Codierung zusätzlicher Informationen genutzt werden, die entsprechenden Codewörter können als Codeverletzungen betrachtet werden. Die Codewörter werden so gewählt, dass nie mehr als zwei Nullen nacheinander auftreten. Zusätzlich wird eine NRZI-Codierung (Non-Return to Zero Invert) eingesetzt. Diese codiert

eine Null mit demselben Spannungspegel wie das vorhergehende Bit, bei einer Eins wird der Spannungspegel invertiert. Insgesamt erhält man ein Leitungssignal mit zahlreichen Übergängen (gute Taktregenerierung) und einer gegenüber der NRZ-Codierung nur um 25 % höheren Bandbreite. Weitere Codes (8B10B, 8B6T, MLT-3) werden in /2.28/ beschrieben.

Für eine exakte Bewertung der verschiedenen Codierungen müssen deren Frequenzspektren und Bitfehlerraten in Abhängigkeit des Signal-Rausch-Abstands analysiert werden /2.29/. Eine grobe Bewertung lässt sich wie folgt durchführen: Da die belegte Bandbreite mit der Häufigkeit der Signalwechsel steigt, sind die Manchester-Codes bezüglich der Bandbreiteneffizienz ungünstig, der NRZ- und AMI-Code sind günstig, der RZ-Code liegt dazwischen. Bezüglich der Taktregenerierbarkeit gilt die umgekehrte Reihenfolge, da das Auftreten vieler Signalübergänge zu einem ausgeprägten Anteil der Taktfrequenz im Signalspektrum führt. Bezüglich der Gleichspannungsfreiheit sind die Codes Manchester und AMI günstig.

2.3.3 Kanalcodierung

Ziel der **Kanalcodierung** (*channel coding*) ist es, die Kommunikation gegen Übertragungsfehler zu sichern. Dazu werden die Daten als Codewörter codiert, die den Eigenschaften des Übertragungskanal angepasst sind. Grundsätzlich kann die Kanalcodierung **fehlererkennende** oder **fehlerkorrigierende Codes** verwenden (*error detection* bzw. *error correction*) (→ Abschnitt 2.8). Damit Fehler erkannt bzw. korrigiert werden können, müssen die Nutzdaten durch **redundante** Daten ergänzt werden, die aus den Nutzdaten abgeleitet sind und zusammen mit diesen übertragen werden. Der Empfänger kann dann die von ihm empfangenen, redundanten Daten mit denen vergleichen, die er selbst aus den empfangenen Nutzdaten berechnet hat. Wenn er dabei Unterschiede feststellt, kann er davon ausgehen, dass Übertragungsfehler aufgetreten sind.

2.3.4 Quellencodierung/Datenkompression

Bei der Kanalcodierung wird die zu übertragende Datenmenge vergrößert. Bei der **Datenkompression** (*source coding, data compression*) verfolgt hingegen das Ziel, die Datenmenge zu **reduzieren**. Verfahren zur Datenkompression lassen sich einteilen in Entropiecodierung, Quellencodierung und hybride Codierung. Die **Entropiecodierung** ist verlustfrei, d. h. die Originaldaten können exakt wiederhergestellt werden. Dazu entfernt der Sender die im Signal enthaltene **Redundanz**, der Empfänger fügt sie

wieder hinzu. Beispielsweise kann eine Folge n gleicher Zeichen durch ein Zeichen und den Wiederholungsfaktor n übertragen werden. Die **Quellencodierung** verwendet Wissen über die zu codierenden Signale. Sie entfernt die im Signal enthaltene **Irrelevanz** (dies sind Anteile, die der Empfänger nicht oder nur schlecht wahrnehmen kann, z. B. Farbverläufe bei rasch veränderlichen Kanten in Bildern). Die Quellencodierung ist meistens verlustbehaftet, erreicht aber dafür wesentlich höhere Kompressionsgrade als die Entropiecodierung.

Hybride Codierungsverfahren sind meistens Kombinationen aus Entropie- und Quellencodierung. Bild 2.13 ordnet einige bekannte Verfahren in dieses Schema ein, weitere Information enthalten /2.12/ und /2.31/.

Entropiecodierung

Laufängencodierung: Folge gleicher Symbole wird durch ein Symbol und die Wiederholungsanzahl ersetzt

Huffman-Codierung: häufig vorkommende Symbole erhalten kurze Codewörter; reduziert die mittlere Länge der zu übertragenden Codewörter

Arithmetische Codierung: Darstellung von Nachrichten als Intervalle auf der Achse rationaler Zahlen

Quellencodierung (Prädiktion)

DPCM (Differential Pulse Code Modulation): die Differenz aufeinander folgender Codewörter wird übertragen; reduziert die Anzahl Bits pro Wort

DM (Delta Modulation): wie DPCM, jedoch Codierung der Differenz in einem Bit

Quellencodierung (Transformation)

FFT (Fast Fourier Transform): Berechnung des Frequenzspektrums, Spektralanteile geringer Amplitude werden nicht übertragen

DCT (Discrete Cosine Transform): vereinfachte Form der FFT; verwendet bei JPEG, MPEG

Quellencodierung (Vektorquantisierung)

Gruppen von Signalwerten (Vektoren) werden durch ein Codewort variabler Länge repräsentiert. Kurze Codewörter für häufig auftretende Vektorwerte

Bild 2.13 Übersicht Datenkompression
FFT (Fast Fourier Transform)
JPEG (Joint Photographic Experts Group)
MPEG (Motion/Moving Picture Experts Group)

Hybride Codierung

JPEG (Joint Photographic Experts Group): für Standbilder. Blöcke werden mit DCT codiert, Codierung der Spektralanteile mit unterschiedlicher Auflösung, anschließend eine Entropiecodierung.

JPEG 2000: erhöhte Kompression bzw. Qualität. Verwendet Wavelet Transform und arithmetische Codierung.

MPEG (Motion Picture Experts Group): für Bewegtbilder. Redundanz und Irrelevanz innerhalb einzelner Frames (ein Bild) wie auch zwischen aufeinander folgenden Frames werden reduziert. Asymmetrie: hoher (geringer) Aufwand für Codierung (Decodierung), MPEG 1 und MPEG 2 nutzen DCT, MPEG 4 verwendet u. a. Wavelets.

Bild 2.13 Übersicht Datenkompression (Fortsetzung)