

Thomas Beier
Tobias Kaufhold
Mohammadreza Paydar



Messdatenverarbeitung mit LabVIEW und Python



2., vollständig überarbeitete Auflage

HANSER

Beier / Kaufhold / Paydar
**Messdatenverarbeitung mit
LabVIEW und Python**



Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie dazu einfach diesen Code ein:

plus-t8vfn-88bx4

plus.hanser-fachbuch.de



Bleiben Sie auf dem Laufenden!

Hanser Newsletter informieren Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der Technik. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter

www.hanser-fachbuch.de/newsletter

Thomas Beier
Tobias Kaufhold
Mohammadreza Paydar

Messdatenverarbeitung mit LabVIEW und Python

2., vollständig überarbeitete Auflage

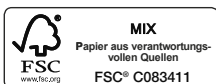
HANSER

Über die Autor:innen:

Thomas Beier unterrichtete an der Staatlichen Technikerschule Berlin die Fächer Messdatenverarbeitung und Automatisierungstechnik.

Tobias Kaufhold ist als Lehrkraft an der Staatlichen Technikerschule Berlin tätig.

Dipl.-Ing. Mohammadreza Paydar unterrichtet zahlreiche Fächer im Bereich Elektrotechnik an der Staatlichen Technikerschule Berlin.



Print-ISBN: 978-3-446-48248-7

E-Book-ISBN: 978-3-446-48435-1

Die allgemein verwendeten Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Alle in diesem Werk enthaltenen Informationen, Verfahren und Darstellungen wurden zum Zeitpunkt der Veröffentlichung nach bestem Wissen zusammengestellt. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Werk enthaltenen Informationen für Autor:innen, Herausgeber:innen und Verlag mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor:innen, Herausgeber:innen und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Weise aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso wenig übernehmen Autor:innen, Herausgeber:innen und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt also auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die endgültige Entscheidung über die Eignung der Informationen für die vorgesehene Verwendung in einer bestimmten Anwendung liegt in der alleinigen Verantwortung des Nutzers.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet unter <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Werkes, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Einwilligung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung – mit Ausnahme der in den §§ 53, 54 UrhG genannten Sonderfälle –, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Wir behalten uns auch eine Nutzung des Werks für Zwecke des Text und Data Mining nach § 44b UrhG ausdrücklich vor.

© 2025 Carl Hanser Verlag GmbH & Co. KG, München
Vilshofener Straße 10 | 81679 München | info@hanser.de
www.hanser-fachbuch.de

Lektorat: Dipl.-Ing. Natalia Silakova-Herzberg

Herstellung: le-tex publishing services GmbH, Leipzig

Coverkonzept: Marc Müller-Bremer, www.rebranding.de, München

Covergestaltung: Max Kostopoulos

Titelmotiv: © Fotolia / Wladimir Bulgar

Satz: Eberl & Koesel Studio, Kempten

Druck: CPI Books GmbH, Leck

Printed in Germany

Vorwort

Das Fach Messdatenverarbeitung, das nun unter dem neuen Namen Signale und Systeme an der Staatlichen Technikerschule Berlin (STB) unterrichtet wird, hat eine lange Tradition an der STB. Es umfasst weiterhin den zusätzlichen Schwerpunkt der digitalen Signalverarbeitung. Die Inhalte werden praxisnah mit Messgeräten und unter Einsatz der Programmiersprache LabVIEW vermittelt. Die erste Auflage dieses Buches ist im Wesentlichen aus dem Unterrichtsmaterial von Thomas Mederer entstanden, der das Fach über viele Jahre an der STB lehrte und Mitautor der ersten Auflage war. Ihm sei an dieser Stelle herzlich gedankt. Seit der Veröffentlichung der ersten Auflage sind mittlerweile zehn Jahre vergangen, in denen zahlreiche Neuerungen eingeflossen sind. Dieses Buch ist weiterhin an der Staatlichen Technikerschule Berlin verankert, und zwei neue Dozenten der Schule sind als Autoren in dieses Projekt eingestiegen.

Das Problem, die anspruchsvolle Theorie mit der Praxis in Einklang zu bringen, bestand von Anfang an. Während der praktische Teil sich gut mit LabVIEW programmieren lässt, ist der theoretische Teil der digitalen Signalverarbeitung mit der vorhandenen Literatur nur schwer zu vermitteln. Meist sind die guten Bücher der digitalen Signalverarbeitung für den mathematisch nicht vorgebildeten Studierenden nur schwer verständlich. Die STB hat im Unterricht von Anbeginn die anschauliche Beschreibung der streng mathematischen Vorgehensweise vorgezogen. Das wurde in der zweiten Auflage beibehalten. Die zweite Auflage wurde gründlich überarbeitet. Das Kapitel digitale Signalprozessoren wurden herausgenommen und stattdessen wurde die Messdatenverarbeitung mit Python aufgenommen. Die Programmiersprache Python hat sich zu einem mächtigen Werkzeug in vielen Bereichen entwickelt. LabVIEW und Python ergänzen sich zusammen hervorragend. Über den sogenannten Python-Knoten lässt sich eine direkte Schnittstelle zwischen beiden Programmiersprachen realisieren.

Mit diesem Buch haben sich die Autoren zum Ziel gesetzt, den Studierenden einen leichteren Zugang zur Theorie zu ermöglichen. Die Autoren sind sich bewusst, dass an einigen Stellen die Aussagen zugunsten der Verständlichkeit vereinfacht wurden. Viele allgemeingültige Aussagen, die quasi nur in der Praxis so vorkommen, werden nicht weiter infrage gestellt. An dieser Stelle muss der interessierte Leser auf die weiterführende Literatur zurückgreifen.

Die Inhalte richten sich an Techniker, angehende Studenten, Praktiker und Autodidakten. Mit diesem Buch kann altes Wissen aufgefrischt werden, ohne in die tiefere Mathematik einsteigen zu müssen. Die Lösungen zu den Aufgaben sind teilweise im Buch direkt angegeben und können einfach nachvollzogen werden, sie können auch von der Webseite *plus.hanser-fachbuch.de* heruntergeladen werden. Die Programme, die im Buch teilweise nur auszugsweise dargestellt sind, sind dort komplett vorhanden.

Bei den Lösungsbeispielen wurden im Wesentlichen elementare LabVIEW-Funktionen verwendet und auf die Anwendung fertiger Funktionen verzichtet. Der Programmierstil wurde zeitgemäß angepasst. Die Themen des Buches entsprechen zu größten Teil den Inhalten, die an der Staatlichen Technikerschule Berlin vermittelt werden. Es wurden aber auch viele neue Themen mit aufgenommen.

Wir möchten uns an dieser Stelle für die gute Betreuung durch das Team des Hanser Verlags Frau Kubiak und Frau Silakova bedanken.

Das Autoren-Team wünscht den Lesern viel Freude beim Studium dieses Buches.

Berlin, im August 2025

Thomas Beier

Tobias Kaufhold

Mohammadreza Paydar

Inhalt

Vorwort	V
1 Einführung	1
1.1 Historische Entwicklung	1
1.2 Moderne Messwerterfassung und -verarbeitung	4
2 Analog-Digital-Umsetzer	7
2.1 Grundbegriffe und Kennlinie	7
2.2 Verfahren zur AD-Umsetzung	10
2.2.1 Sukzessive Verfahren	10
2.2.2 Sigma-Delta-Verfahren	12
2.2.3 ADU-Verfahren in der Übersicht	15
2.3 Quantisierungsfehler	15
2.4 Die effektive Anzahl von Bits (ENOB)	18
2.5 Codierungen für Datenwandler	19
2.6 AD-Umsetzung bei zeitveränderlicher Eingangsspannung	22
2.7 Sample-Hold-Schaltung	23
2.8 Nichtlineare Quantisierung	28
2.8.1 μ -Law und A-Law	28
2.8.2 μ -Law-Kompandierung	29
2.9 Adaptive Differential Pulse Code Modulation (ADPCM)	31
2.10 Übungen	31
2.11 Lösungen	33

3	Digital-Analog-Umsetzer	37
3.1	Grundbegriffe und Kennlinie	37
3.2	Verfahren zur DA-Umsetzung	40
3.3	Fehlerbetrachtung	41
3.4	Übungen	45
3.5	Lösungen	45
4	Signalabtastung und Signalrekonstruktion	49
4.1	Abtastung und Abtast-Theorem	49
4.2	Rekonstruktion des analogen Signals	53
4.3	Übungen	56
4.4	Lösungen	57
5	Messwerterfassungssysteme	59
5.1	Grundstrukturen von Messwerterfassungssystemen	59
5.2	Ausgewählte Sensoren	65
5.2.1	Dehnmessstreifen (DMS)	65
5.2.2	Kapazitive Messfühler	67
5.2.3	Induktive Messfühler	75
5.3	Signalkonditionierung	77
5.4	Schaltungen zur Signalkonditionierung	78
5.4.1	Messverstärker	79
5.4.2	Filter	96
5.4.2.1	Aktive Filter erste Ordnung	96
5.4.2.2	Butterworth-Filter	103
5.4.2.3	Tschebyscheff-Filter	103
5.4.2.4	Bessel-Filter	104
5.4.2.5	Sallen-Key-Schaltung	104
5.4.2.6	Kaskadierung von aktiven Filtern	108
5.4.3	Messbrücken	113
5.4.4	Trägerfrequenz-Messverfahren	115
5.5	Übungen	120
5.6	Lösungen	122

6	Grundlagen zur digitalen Signalverarbeitung	127
6.1	Einführung	127
6.2	Grundelemente der digitalen Signalverarbeitung	137
6.3	Testsignale	144
6.4	Übungen	147
6.5	Lösungen	148
7	Digitale Filter	153
7.1	Filterentwurf	156
7.2	Filterentwurf mit einem Toleranzschema	157
7.3	IIR-Filterentwurf mit der impulsinvarianten z-Transformation	161
7.4	Filterentwurf mit der bilinearen z-Transformation	164
7.5	Transformation zwischen Tiefpass- und Hochpassfilter	172
7.6	Realisierung von Bandsperren und Bandpässen	174
7.7	Filter höherer Ordnung	176
7.8	Entwurfsverfahren für FIR-Filter	181
7.9	FIR-Filterentwurf mit einem Approximationsverfahren	184
7.10	Savitzky-Golay-Filter	195
7.11	Kalman-Filter	202
	7.11.1 Mathematische Modellierung	202
	7.11.2 Mathematisches Modell des Zustandsraums eines Beobachters (Kalman-Filter)	205
	7.11.3 Implementierung in LabVIEW	216
7.12	Übungen	218
7.13	Lösungen	219
8	Diskrete Fourier-Transformation	223
8.1	Übungen	236
8.2	Lösungen	237
9	Messdatenverarbeitung mit LabVIEW	243
9.1	Datenerfassung	243
9.2	Software für die Messdatenverarbeitung	246
9.3	Der VISA-Treiber	247
9.4	Aufgabe: Amplitudengang eines Filters	251
9.5	Aufgabe: Waage mit Dehnungsmessstreifen	259

9.6	Programmierung mit NI-USB-Messmodulen	263
9.6.1	Hardwareaufbau von Messmodulen	263
9.6.2	Einrichtung von NI-Hardware mit dem NI MAX	265
9.6.3	Verwendete LabVIEW-Funktionen	268
9.7	Aufgabe: Temperaturmessung	269
9.8	Aufgabe: Wetterstation für Druck, Feuchte und Temperatur	272
9.9	Aufgabe: Signalanalyse eines Sinussignals	274
9.10	Aufgabe: Signalanalyse eines Mikrofonsignals	276
9.11	Aufgabe: Digitale Signalverarbeitung mit Musik	279
9.12	Aufgabe: Analyse eines EKG-Signals	281
9.13	Aufgabe: Template für Mess- und Automatisierungsaufgaben	283
10	Messdatenverarbeitung mit Python	287
10.1	Verwendung von NI-DAQmx in Python	287
10.1.1	NI-DAQmx-Integration in Python: Software installieren	289
10.1.2	Grundlegende Verwendung von NI-DAQmx in Python	291
10.1.3	Erzeugen und Ausgeben einer Sinuswelle mit NI-DAQmx in Python	292
10.1.4	Temperaturerfassung mit LM35 und NI USB-6009	293
10.2	Regressionsanalyse	297
10.2.1	Lineare Regression	297
10.2.2	Ausgleichspolynom	303
10.2.3	Lineare Regression und Differenzialgleichungen	306
10.2.4	Linearisierung	310
10.3	Python in der Bildverarbeitung und OpenCV	312
10.3.1	Echtzeit-Objektmaskierung mit OpenCV	312
10.3.2	Objekterkennung mit YOLO (You Only Look Once)	313
10.4	Integration von Python-Code in LabVIEW	318
10.5	Übungen	322
10.6	Lösungen	323
	Quellenverzeichnis	327
	Index	329

1

Einführung

Messen gehört wahrscheinlich zu den ältesten Fähigkeiten der Menschheit – und ist heute wichtiger denn je. Schon in der Steinzeit mussten Länge und Masse bestimmt werden, damit der Warenaustausch fair blieb. Mangels moderner Messmittel griff man damals kurzerhand auf das zurück, was man immer dabei hatte: den eigenen Körper. So wurden die Körpermaße des Menschen wie z.B. Fuß oder Elle die ersten Maßverkörperungen der Geschichte.

Heute kann deutlich präziser gemessen werden – und auch digitaler. Doch egal ob damals oder heute: Messwerte allein nützen wenig, wenn sie nicht sinnvoll verarbeitet werden können. Messdatenverarbeitung sorgt hier dafür, dass rohe Messwerte geordnet, übertragen, umgeformt und letztlich in nützliche Informationen verwandelt werden, im Allgemeinen mithilfe von Computern oder Mikrocontrollern.

Wie rasant sich dieser Bereich entwickelt hat, zeigt ein kleiner geschichtlicher Rückblick. Denn eines ist sicher: Ohne Messdatenverarbeitung wäre unsere technisierte Welt undenkbar.

1.1 Historische Entwicklung

Messinstrumente vor Strom und Zeiger

Bevor Elektrotechnik die Messtechnik revolutionierte bediente man sich mechanischer, optischer oder sogar akustischer Methoden um die Welt zu erfassen.

Massen wurden mit einfachen Balkenwaagen bestimmt, Längen mit Maßstäben, die Zeit ließ sich mithilfe von Wasseruhren oder Sonnenuhren grob erfassen, frühe Thermometer und Barometer arbeiteten mit Flüssigkeiten und Luftdruck, ganz ohne Elektrizität. Selbst Frequenzen ließen sich durch Stimmgabeln nachweisen. Diese frühen Messmittel bildeten das Fundament der Messtechnik.

Elektromechanische Instrumente

Bevor digitale Sensoren und Mikrocontroller die Bühne betraten, entwickelte sich gegen Ende des 19. Jahrhunderts die elektromechanische Messtechnik. Eine wichtige Gruppe der elektromechanischen Instrumente waren die Zeigerinstrumente. Bei ihnen wurde elektrische Größen wie Spannung oder Strom raffiniert in eine mechanische umgewandelt und somit sichtbar gemacht. Wichtige Vertreter solcher Instrumente waren das Drehspul- und das Dreheiseninstrument, aber auch frühe Oszillografen wie z.B. das Lichtstrahl-Oszilloskop. Das Messprinzip des Drehspul- oder Dreheiseninstruments spielt auch noch heute bei den digitalen Messinstrumenten eine Rolle, wenn z.B. der wahre Effektivwert (RMS) ermittelt werden soll.

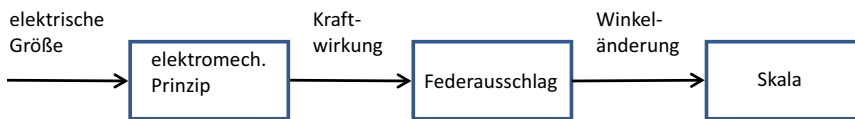


Bild 1.1 Elektromechanische Wirkkette

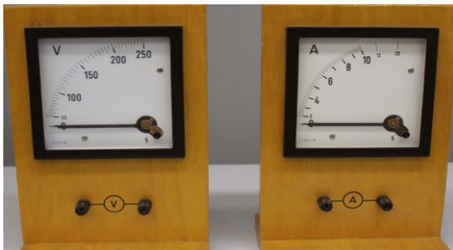


Bild 1.2
Elektromechanische Voltmeter

Elektronische Instrumente

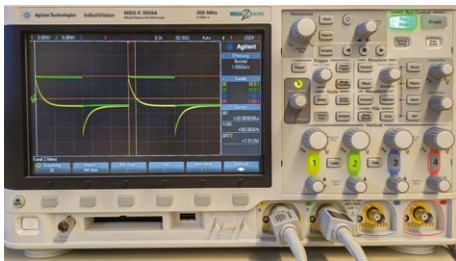
Seit dem 19. Jahrhundert wird in Verbindung mit Elektronenröhren von Elektronik gesprochen, eine neue Ära der Messtechnik begann. Elektronische Bauelemente ersetzen mechanische und erweiterten nicht nur den Einsatzbereich konventioneller Instrumente, sondern ermöglichten auch neue Messmethoden. So finden elektronische Hilfsmittel Anwendung in der Sensorik, bei Messverstärkern, in der Signalaufbereitung und -erzeugung sowie in der Übertragung von Messdaten. Wichtige Messgeräte dieser Art waren z.B. das Elektronenstrahl-Oszilloskop. Plötzlich konnten Daten nicht nur angezeigt, sondern auch gespeichert, verarbeitet werden. Der Grundstein für die heutige digitale Messtechnik war gelegt.

**Bild 1.3**

Elektronenstrahl-Oszilloskop

Digitale Instrumente

Mit der Verfügbarkeit integrierter Schaltungen zur Analog-Digital-Umsetzung (ADU) und Digital-Analog-Umsetzung (DAU) begann sich in den 1970er-Jahren die digitale Messtechnik durchzusetzen. Diese Entwicklung ging einher mit zunehmender Verbreitung der digitalen Signalverarbeitung, wodurch Messwerte nicht nur präziser erfasst, sondern auch schneller verarbeitet und analysiert werden konnten. Geräte der digitalen Messtechnik sind z.B. digitale Multimeter (DMM), Zähler (Counter) und das digitale Speicheroszilloskop (DSO).

**Bild 1.4**

Digitales Speicheroszilloskop

Rechnergestützte Messtechnik

Mit der steigenden Rechenleistung von Computern hat sich auch die Messtechnik grundlegend gewandelt. Heute verfügen viele Messgeräte über eigene eingebaute Mikrorechner, die eine schnelle und präzise Datenverarbeitung ermöglichen. Messwerterfassungs-Hardware ist in vielfacher Weise über geeignete Schnittstellen oder Bussysteme mit Rechnern unterschiedlichster Art verbunden bzw. vernetzt. Auch die zugehörige Software entwickelte sich von Assemblersprachen über Hochsprachen wie FORTRAN, BASIC und C hin zu komplexen, grafisch orientierten Programmierumgebungen, wie z.B. LabVIEW.

Eine immer wichtigere Rolle spielt dabei auch Python. Die Sprache hat sich in der Messtechnik durch ihre einfache Syntax, die breite Unterstützung für wissenschaftliches Rechnen und die riesige Auswahl an Bibliotheken etabliert.



Bild 1.5
USB-6001 Messbox von NI

1.2 Moderne Messwerterfassung und -verarbeitung

Moderne Messtechnik ist in nahezu allen technischen und wissenschaftlichen Bereichen unverzichtbar. Ihre Anwendungen reichen von Aufgaben in der Mess- und Regelungstechnik über die Datenübertragung bis hin zur Sprach- und Bildverarbeitung. Wichtige Anwendungsgebiete finden sich in der Automatisierung, in der Medizintechnik oder in der Kommunikationstechnologie.

Für Messwerterfassungssysteme (engl. DAQ system = Data-Acquisition-System) ergeben sich aus den Messaufgaben die unterschiedlichsten Anforderungen. Bei der Konzipierung von Messsystemen stellen sich daher zentrale Fragen, darunter:

- Welche Messgrößen sollen erfasst werden – analog oder digital?
- Ist Signalkonditionierung erforderlich?
- Müssen mehrere Messgrößen zeitgleich erfasst werden?
- Welche Bandbreite an Frequenzen muss erfasst werden?
- Welche Messunsicherheit ist zulässig?
- Wie lange soll die Messung dauern?
- Ist eine Echtzeit-Berechnung erforderlich?

Diese Faktoren bestimmen maßgeblich die Auswahl der geeigneten Messhardware und -software.

Für viele Anwendungen lässt sich die in Bild 1.6 dargestellte Grundstruktur für die Messwerterfassung und -verarbeitung skizzieren. Die Messkette besteht aus mehreren Schritten, die gemeinsam dafür sorgen, dass eine physikalische Größe erfasst, verarbeitet und ausgegeben wird.

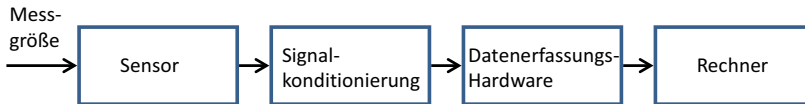


Bild 1.6 Grundstruktur eines Messwerterfassungssystems

Die einzelnen Funktionseinheiten seien im Folgenden kurz erläutert:

Sensor

Der Sensor (auch Messfühler oder Messwertaufnehmer) reagiert auf die nichtelektrische Messgröße. Es werden dabei physikalische und chemische Effekte genutzt, um die nichtelektrische Größe in eine elektrische Größe umzuwandeln.

Signalkonditionierung

Das vom Sensor erzeugte Signal ist oft schwach oder nicht direkt nutzbar. Die Signalkonditionierung bereitet das Ausgangssignal des Sensors so auf, dass es für die weitere Verarbeitung geeignet ist. Ist das Ausgangssignal der Signalkonditionierung ein sog. Einheitssignal, dann wird die Signalkonditionierung auch als Messumformer oder Transmitter (engl. transducer) bezeichnet. Die Signalkonditionierung enthält elektronische Komponenten wie zum Beispiel Verstärker oder Filter.

Datenerfassungs-Hardware

In der Datenerfassungs-Hardware werden die ankommenden Signale digitalisiert und für die digitale Weiterverarbeitung umgeformt. Typische Komponenten der Datenerfassungs-Hardware sind Multiplexer, Verstärker, AD-Wandler und Mikroprozessoren. Die Hardware bildet die Schnittstelle zwischen der analogen Welt und der digitalen Verarbeitung.

Rechner

Die digitalisierten Messwerte können zur weiteren Verarbeitung an einen Rechner übertragen werden. Je nach Anwendungsfall können unterschiedliche Rechner wie z.B. PC, Mikrocontroller oder Digitaler Signalprozessor mit entsprechender Peripherie zum Einsatz kommen. Der Rechner analysiert die Daten, stellt sie visuell dar oder leitet sie zur weiteren Nutzung weiter.

Die einzelnen Funktionseinheiten können dabei sowohl als einzelne Bausteine vorliegen oder in einer einzigen integrierten Lösung zusammengefasst sein – wie bei einem „Smart-Sensor“, einem Sensor, der Signalverarbeitung und Datenausgabe in einem Chip vereint. Durch den stetigen technologischen Fortschritt entwickeln sich sowohl Hard- als auch Software für Messwerterfassungssysteme kontinuierlich weiter und ermöglichen immer leistungsfähigere Lösungen.

2

Analog-Digital-Umsetzer

Ein Analog-Digital-Umsetzer (ADU), auch Analog-Digital-Wandler oder Analog-Digital-Converter (ADC) genannt, setzt eine analoge Eingangsspannung in eine Zahl um.

2.1 Grundbegriffe und Kennlinie

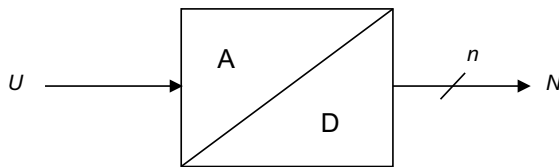


Bild 2.1
Blockbild AD-Umsetzer

Bild 2.1 zeigt das Blockbild für einen AD-Umsetzer. Der Bereich der Eingangsspannung U wird über eine Referenzspannung U_{ref} festgelegt. Die analoge Eingangsgröße kann jeden möglichen Zwischenwert im Spannungsbereich annehmen. Die Zahl N als Ausgangsgröße kann nur diskrete Werte annehmen. Sie wird meist als Dualzahl mit n Bits ausgegeben.

Folgende Gleichung, die **Wandlungsfunktion**, beschreibt das Übertragungsverhalten:

$$N = \left\langle \frac{U}{U_{\text{LSB}}} \right\rangle \quad (2.1)$$

mit

$$U_{\text{LSB}} = \frac{U_{\text{ref}}}{2^n}$$

Die „eckige Klammer“ bedeutet hierbei, dass N gerundet wird und nur ganze Zahlen annehmen kann.

U_{ref} = Referenzspannung, legt den Bereich der Eingangsspannung U fest

U_{LSB} = Quantisierungseinheit = Spannungsstufung der Eingangsspannung U



Hinweis

Die Ausgangsgröße N eines ADU gibt an, wie oft die Quantisierungseinheit U_{LSB} in der Eingangsspannung U gerundet enthalten ist.

Die grafische Darstellung der Wandlungsfunktion ergibt die **Kennlinie eines ADU**.



Beispiel 2.1

Für einen 3-Bit-ADU ($n = 3$) mit $U_{\text{ref}} = 4 \text{ V}$ ist die Kennlinie zu zeichnen.

Lösung

$$U_{\text{LSB}} = \frac{U_{\text{ref}}}{2^n} = \frac{4 \text{ V}}{2^3} = 0,5 \text{ V}$$

Die sich damit nach Formel 2.1 ergebende Kennlinie ist in Bild 2.2 dargestellt.

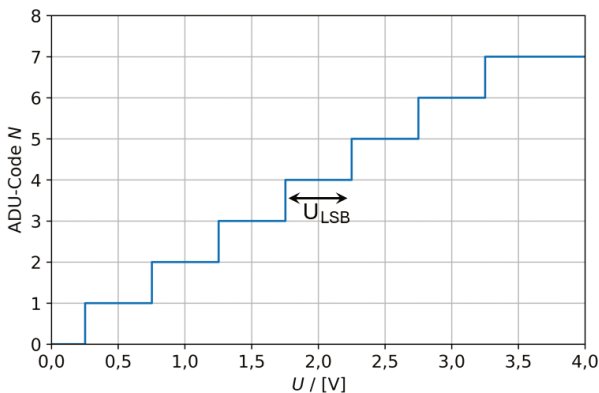


Bild 2.2 ADU-Kennlinie

Quantisierungsfehler

Da die AD-Wandlung unendlich viele analoge Spannungswerte auf eine endliche Anzahl von Zahlenwerten abbildet (dieser Vorgang wird auch als Quantisierung bezeichnet), lässt sich von einem quantisierten Zahlenwert N nicht mehr eindeutig auf die Eingangsspannung U zurückschließen. Es entsteht eine Unsicherheit, die als „Quantisierungsfehler“ bezeichnet werden kann. Der Quantisierungsfehler ist ein systematischer Fehler, der bei jeder AD-Umsetzung entsteht. Er kann durch höhere Auflösung des ADU (höhere Bit-Anzahl n) hinreichend klein gemacht werden.

Bei vorliegendem Zahlenwert N lässt sich bezüglich der gewandelten Spannung U nur noch folgende Aussage machen:

$$U = N \cdot U_{\text{LSB}} \pm \frac{1}{2} \cdot U_{\text{LSB}} \quad (2.2)$$

Die Unsicherheit $\pm \frac{1}{2} U_{\text{LSB}}$ drückt den entstehenden Quantisierungsfehler aus. Im Beispiel (Bild 2.2) lässt sich das unmittelbar nachvollziehen.:

Für z.B. $N = 3$ ergibt sich:

$$U = N \cdot U_{\text{LSB}} \pm \frac{1}{2} \cdot U_{\text{LSB}} = 3 \cdot 0,5 \text{ V} \pm \frac{1}{2} \cdot 0,5 \text{ V} = 1,5 \text{ V} \pm 0,25 \text{ V}$$

d.h. die Eingangsspannung U lag im Bereich 1,25 V bis 1,75 V.



Beispiel 2.2

Zur digitalen Temperaturmessung kommt die in Bild 2.3 dargestellte Schaltung zum Einsatz. Der zu erfassende Temperaturbereich von 10 °C bis 40 °C wird durch den Sensor und die nachfolgende Signalkonditionierung auf einen Spannungsbereich von 0 V bis 10 V abgebildet. Zu ermitteln ist die gemessene Temperatur θ , wenn der AD-Wandler einen Zahlenwert von $N = 12$ ausgibt.

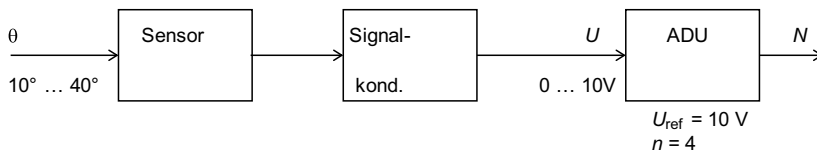


Bild 2.3 Digitale Temperaturmessung

Lösung

Aus den AD-Wandler-Daten $U_{\text{ref}} = 10 \text{ V}$ und $n = 4$ ergibt sich für die Quantisierungseinheit U_{LSB}

$$U_{\text{LSB}} = \frac{10 \text{ V}}{2^4} = 0,625 \text{ V}$$

Für die Eingangsspannung U am AD-Wandler kann demnach angegeben werden:

$$U = N \cdot U_{\text{LSB}} \pm \frac{1}{2} \cdot U_{\text{LSB}} = 12 \cdot 0,625 \text{ V} \pm \frac{1}{2} \cdot 0,625 \text{ V} = 7,7 \text{ V} \pm 0,3125 \text{ V}$$

Zwischen Temperatur und Spannung gilt aufgrund der Abbildung die Relation:

$$\frac{\Delta \theta}{\Delta U} = \frac{30^\circ \text{C}}{10 \text{ V}} = 3 \frac{^\circ \text{C}}{\text{V}}$$

Für die gemessene Temperatur unter Berücksichtigung des Anfangswertes von 10 °C ergibt sich:

$$\theta = 10^\circ \text{C} + U \cdot 3 \frac{^\circ \text{C}}{\text{V}} = 10^\circ \text{C} + 22,5^\circ \text{C} \pm 0,94^\circ \text{C}$$

Die Temperatur würde also bei dieser Messeinrichtung infolge des Quantisierungsfehlers mit einer Ungenauigkeit von $\pm 0,94^\circ\text{C}$ erfasst werden. Bei Verwendung eines 8-Bit-ADU würde die Ungenauigkeit nur noch $\pm 0,06^\circ\text{C}$ betragen.

2.2 Verfahren zur AD-Umsetzung

Ähnlich wie bei den DA-Wandlern gibt es auch hier eine Reihe unterschiedlicher Verfahren, die sich in ihren Eigenschaften wie Umsetzungsgeschwindigkeit, Auflösung, Genauigkeit, Linearität usw. unterscheiden. Beispielhaft sollen zwei Verfahren genauer betrachtet werden.

2.2.1 Sukzessive Verfahren

Das sukzessive Verfahren („schrittweise Annäherung“) entspricht einer Waage, auf der mit entsprechenden Gewichten auf der rechten Seite, das richtige Gewicht auf der linken Seite abgeschätzt wird.

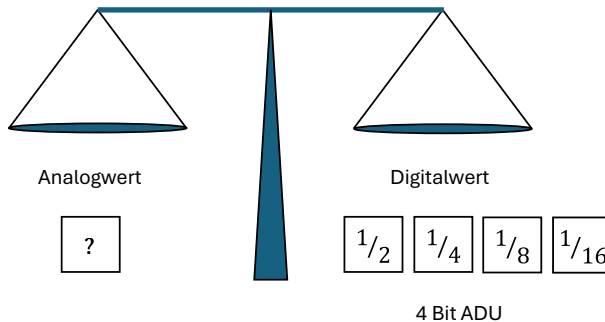


Bild 2.4 Wägeverfahren

Das Bild 2.4 deutet das Verfahren an für einen 4-Bit-Wandler. In Bild 2.5 ist die grundsätzliche Struktur des sukzessiven Verfahrens dargestellt.

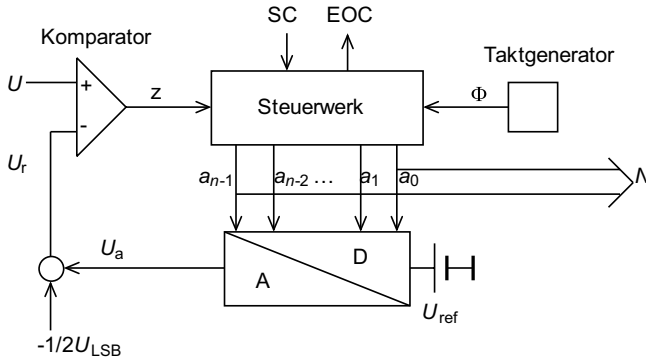


Bild 2.5 Verfahren der sukzessiven Approximation

Die Wandlung einer Eingangsspannung U in eine Zahl N lässt sich anhand von Bild 2.5 folgendermaßen beschreiben:

Nach einem Startimpuls an SC (engl. start conversion) setzt das Steuerwerk die Leitungen a_{n-1} bis a_0 auf null. Damit ist $N = 0$, $U_a = 0\text{ V}$ und die Vergleichsspannung am Komparator $U_r = -1/2 U_{\text{LSB}}$.

Beginnend mit a_{n-1} werden jeweils mit der steigenden Flanke des Taktes Φ die einzelnen Leitungen a_i nacheinander (sukzessive) auf 1 gesetzt. Über den DA-Wandler in der Rückkopplung wird dabei die Vergleichsspannung U_r für den Komparator gemäß $U_r = N \cdot U_{\text{LSB}} - 1/2 U_{\text{LSB}}$ erzeugt. Bei der nachfolgenden fallenden Flanke des Taktes wird dann der Komparatorausgang z durch das Steuerwerk ausgewertet:

- $z = 0$ (d.h. $U_r \geq U$) $\rightarrow a_i = 0$ (Bit wird zurückgesetzt)
- $z = 1$ (d.h. $U_r < U$) $\rightarrow a_i = 1$ (Bit bleibt gesetzt)

Nach der Festlegung für Bit a_0 ist der Vorgang beendet und die Ausgangsgröße N bestimmt. Das Ende der Wandlung wird über das Signal EOC (engl. end of conversion) angezeigt.

In Bild 2.6 ist der zeitliche Verlauf der einzelnen Größen für $U = 5,63\text{ V}$; $U_{\text{ref}} = 8\text{ V}$ und $n = 5$ (Bits a_4 bis a_0) dargestellt. Als Ergebnis wird ein Ausgabewert von $N = 10111_2 = 23$ erzeugt.

Dies korrespondiert mit dem Ergebnis aus der Wandlungsfunktion:

$$U_{\text{LSB}} = \frac{U_{\text{ref}}}{2^n} = \frac{8\text{ V}}{2^5} = 0,25\text{ V}$$

$$N = \left\lfloor \frac{U}{U_{\text{LSB}}} \right\rfloor = \left\lfloor \frac{5,63\text{ V}}{0,25\text{ V}} \right\rfloor = 23$$

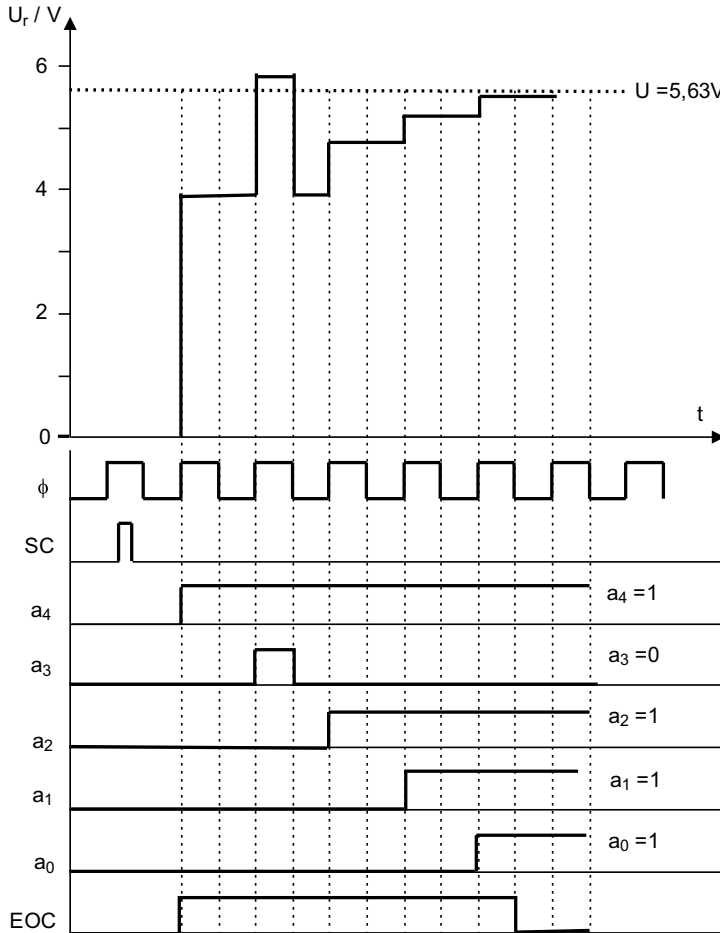


Bild 2.6 Sukzessive Approximation – zeitlicher Ablauf

2.2.2 Sigma-Delta-Verfahren

Das Bild 2.7 zeigt einen einfachen Sigma-Delta-Modulator. Die Eingangsspannung V_{in} wird zunächst mit dem Ausgang eines Rückkopplungs-DAU summiert. Ein Integrator addiert dann den Ausgang dieses Summierungsknotens zu einem Wert, den er aus dem vorherigen Integrationsschritt gespeichert hat. Ein Komparator gibt eine logische 1 aus, wenn der Integrationsausgang größer oder gleich null Volt ist, anderenfalls eine logische 0. Ein 1-Bit-DAC führt den Ausgang des Komparators zurück zum Summierungsknoten: $+U_{ref}$ für logisch 1 und $-U_{ref}$ für logisch 0. Diese Rückkopplung versucht, den Integratorausgang auf null zu halten, indem die Einsen- und Nullen-Ausgabe des Komparators über den DAU rückgekoppelt werden.

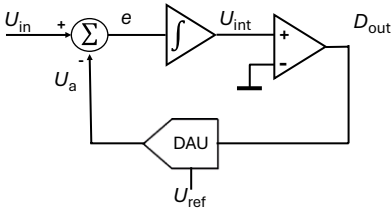


Bild 2.7
Sigma-Delta-Modulator

Der Betrieb des Sigma-Delta-Modulators soll anhand eines Beispiels gezeigt werden.



Beispiel 2.3

Angenommen am Eingang U_{in} liegt eine Spannung 0,5V an. Die Referenzspannung U_{ref} ist 1 V. Der zulässige Eingangsbereich liegt zwischen -1 V und +1 V ($\pm U_{ref}$).

Tabelle 2.1

Schritte	U_{in}	e	U_{int}	D_{out}	U_a	Kommentar
0	0,5 V	0 V	0 V		0 V	Initialisieren
1	0,5 V	0,5 V	0,5 V	1	1 V	$e(1) = U_{in} - U_a(0)$ $U_{int}(1) = U_{int}(0) + e(1)$ $U_{int}(1) \geq 0, D_{out}(1) = 1$ $D_{out}(1) == 1, U_a(1) = 1 \text{ V}$
2	0,5 V	-0,5 V	0 V	1	1 V	$e(2) = U_{in} - U_a(1)$ $U_{int}(2) = U_{int}(1) + e(2)$ $U_{int}(2) \geq 0, D_{out}(2) = 1$ $D_{out}(2) == 1, U_a(2) = 1 \text{ V}$
3	0,5 V	-0,5 V	-0,5 V	0	-1 V	$e(3) = U_{in} - U_a(2)$ $U_{int}(3) = U_{int}(2) + e(3)$ $U_{int}(3) < 0, D_{out}(3) = 0$ $D_{out}(3) == 0, U_a(3) = -1 \text{ V}$

Tabelle 2.1 (Fortsetzung)

Schritte	U_{in}	e	U_{int}	D_{out}	U_a	Kommentar
4	0,5V	1,5 V	1 V	1	1 V	$e(4) = U_{in} - U_a(3)$ $U_{int}(4) = U_{int}(3) + e(4)$ $U_{int}(4) \geq 0, D_{out}(4) = 1$ $D_{out}(4) == 1, U_a(4) = 1 V$
usw.						

Der Bit-Stream ist dann 1101. Von diesen vier Werten sind drei Werte eins. Das entspricht 75 % der vier Werte. Der Spannungsbereich ist durch V_{ref} vorgegeben und beginnt in diesem Beispiel bei $-1 V$. Dann lässt sich der Ausgangswert nach Formel 2.3 berechnen.

$$Y = -U_{ref} + \frac{\text{Anzahl Einsen}}{\text{Gesamtanzahl BitStream}} \cdot 2 \cdot U_{ref}$$

(2.3)

$$Y = -1 V + 0,75 \cdot 2 \cdot 1 V = 0,5 V$$

Das Beispiel ist hier einfach und so gewählt, dass nach vier Schritten das richtige Ergebnis ermittelt wird. Bei einer Eingangsspannung von $V_{in} = 0,7 V$ werden schon mehr Schritte erforderlich, nämlich 20 Schritte. Der Bit-Stream ist dann 1110111110111110111. Das wären 17 Einsen und 3 Nullen. Nach Formel 2.3 ergibt sich dann:

$$Y = -1 V + \frac{17}{20} \cdot 2 \cdot 1 V = 0,7 V$$

Der Strom aus Einsen und Nullen wird nach der Sigma-Delta-Modulationschleife digital gefiltert, siehe Formel 2.3. Die Sigma-Delta-Modulatorschleife läuft typischerweise mit einer viel höheren Frequenz als die endgültige Ausgaberate des digitalen Filters. Beispielsweise kann ein Konverter mit einer Ausgabedatenrate von 2 kHz eine Modulatorschleifenfrequenz von über 2,5 MHz haben. Die Ausgabedatenrate entspricht somit der Sample Rate (Abtastung). Es wird auch deutlich, dass mit zunehmenden Bit-Stream die Auflösung zunimmt.

2.2.3 ADU-Verfahren in der Übersicht

In der Tabelle 2.2 sind wichtige Verfahren zur AD-Umsetzung zusammengefasst.

Tabelle 2.2 ADU-Verfahren

Verfahren	Eigenschaften	Einsatzbeispiele
sukzessive Approximation (Wägeverfahren)	mittlere Wandlungszeit (µs) Auflösung bis 16 Bit	Messkarten, Mikrocontroller
Integrationsverfahren (Dual Slope)	gute Störunterdrückung lange Wandlungszeit (100 ms)	Messinstrumente
Parallelverfahren (Flash-Verfahren)	kurze Wandlungszeit (ns) hoher Schaltungsaufwand	Videotechnik
Sigma-Delta-Verfahren	hohe Auflösung bis 32 Bit, mittlere Wandlungszeit gute Linearität	Audiotechnik, Messkarten

2.3 Quantisierungsfehler

In diesem Abschnitt soll noch einmal etwas ausführlicher auf den schon erwähnten Quantisierungsfehler eingegangen werden. Entsprechend der Fehlerdefinition in der Messtechnik ist er als Differenz zwischen der rückgewandelten Spannung U_{DA} und der ursprünglichen Spannung U definiert (Bild 2.8).

Quantisierungsfehler:

$$F_q = U_{DA} - U = N \cdot U_{LSB} - U$$

(2.4)

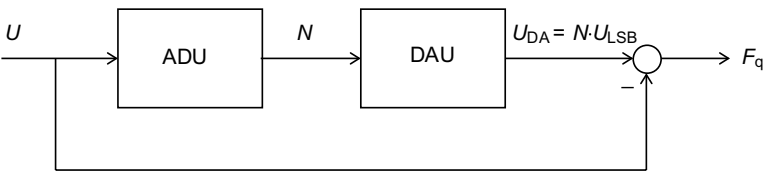


Bild 2.8 Quantisierungsfehler

In Bild 2.9 ist der Verlauf des Quantisierungsfehlers über der Eingangsspannung U des ADU dargestellt. Wie schon in Abschnitt 2.2 festgestellt, bewegt er sich im Bereich $\pm \frac{1}{2}U_{LSB}$.

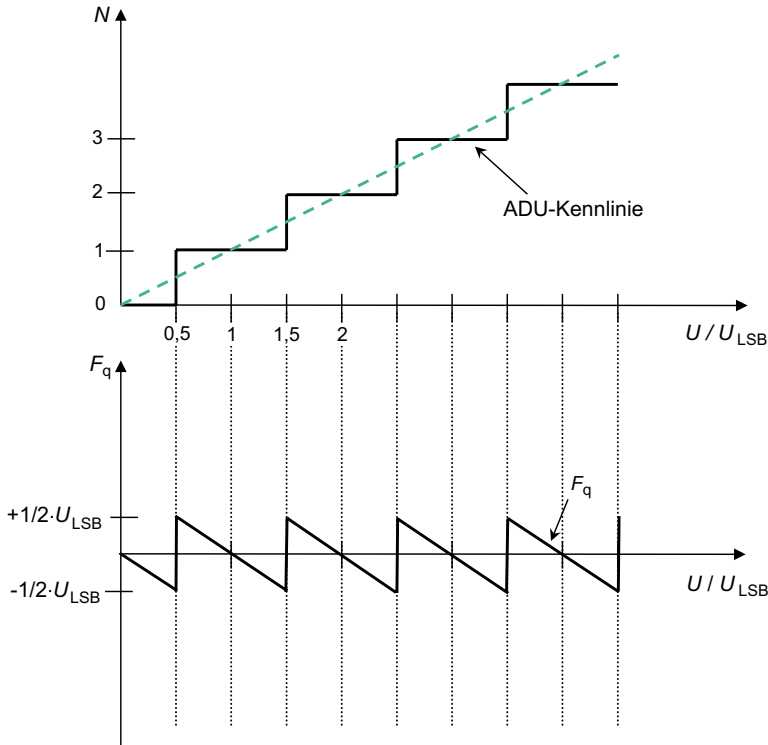


Bild 2.9 Quantisierungsfehler eines idealen ADU

Quantisierungsfehler bei zeitveränderlicher Eingangsspannung

Wird als Eingangsspannung am ADU eine Spannung $u(t)$ betrachtet, die sich zeitlinear ändert, dann lässt sich in Bild 2.9 die Spannungs-Achse (U -Achse) durch eine Zeit-Achse (t -Achse) ersetzen. Der Quantisierungsfehler ergibt sich dann als zeitveränderliche Funktion $F_q(t)$.

$$F_q(t) = u_{\text{DA}}(t) - u(t) = N(t) \cdot U_{\text{LSB}} - u(t) \quad (2.5)$$

Er lässt sich durch seinen Effektivwert beschreiben:

$$F_{q\text{RMS}} = \frac{\hat{F}_q}{\sqrt{3}} = \frac{\frac{1}{2}U_{\text{LSB}}}{\sqrt{3}} = \frac{U_{\text{LSB}}}{\sqrt{12}} \quad (2.6)$$

Der Quantisierungsfehler wird bei zeitveränderlicher Eingangsspannung auch als Quantisierungsrauschen bezeichnet, weil er sich in der rückgewandelten Spannung wie ein überlagertes Rauschen äußert (Bild 2.10). Die Umstellung von Formel 2.5 macht diese Interpretation deutlich:

$$u_{\text{DA}}(t) = u(t) + F_q(t)$$

(rückgewandelte Spannung = Originalspannung + Quantisierungsrauschen)

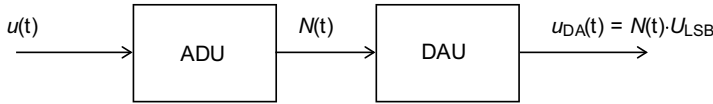


Bild 2.10 Rückgewandelte Spannung und Quantisierungsfehler

Der Einfluss des Quantisierungsrauschens auf das Signal wird durch das Signal-Rausch-Verhältnis SNR (engl. signal to noise ratio) beschrieben:

$$SNR = \frac{U_{RMS}}{F_{qRMS}} \quad (2.7)$$

U_{RMS} = Effektivwert der Signalspannung $u(t)$

F_{qRMS} = Effektivwert des Quantisierungsrauschens $F_q(t)$

Für einen n -Bit-ADU ergibt sich damit bei Vollaussteuerung mit einem Sinussignal $u(t) = U_S \cdot \sin(\omega t)$.

Vollaussteuerung bedeutet: $U_{ref} = 2 \cdot U_S$

mit

$$U_{RMS} = \frac{U_S}{\sqrt{2}} = \frac{U_{ref}}{2 \cdot \sqrt{2}}$$

und

$$F_{qRMS} = \frac{U_{LSB}}{\sqrt{12}} = \frac{U_{ref}}{2^n \cdot \sqrt{12}}$$

folgt

$$SNR = \frac{U_{RMS}}{F_{qRMS}} = \sqrt{1,5} \cdot 2^n \quad (2.8)$$

Üblicherweise wird das Signal-Rausch-Verhältnis in dB (Dezibel) ausgedrückt.

$$SNR_{dB} = 20 \cdot \lg(SNR) = 20 \cdot \lg(\sqrt{1,5} \cdot 2^n) = 20 \cdot \lg \sqrt{1,5} + 20 \cdot n \cdot \lg 2$$

$$SNR_{dB} = (1,76 + n \cdot 6,02) \text{ dB} \quad (2.9)$$

Für einen 8-Bit-ADU ($n = 8$) ergibt sich nach Formel 2.8 bzw. Formel 2.9:

$$SNR = 314 \text{ bzw. } SNR_{dB} = 49,92 \text{ dB}$$

oder anders ausgedrückt: Das Quantisierungsrauschen beträgt 1/314 bzw. 0,03 % des Signals.

Bei realen AD-Umsetzern wird das Signal-Rausch-Verhältnis messtechnisch über eine Spektralanalyse ermittelt. Diese Messung erfasst zusätzlich Verzerrungen (engl. distortion) aufgrund von Nichtlinearitäten in der Kennlinie. Das so bestimmte Signal-Rausch-Verhältnis wird mit $SINAD$ (engl. signal to noise and distortion) bezeichnet, einem Maß für die Gesamtqualität des digitalisierten Signals, das sowohl Rauschen als auch harmonische Verzerrungen durch ADC-Nichtlinearitäten berücksichtigt ($SINAD < SNR$).

2.4 Die effektive Anzahl von Bits (ENOB)

In realen Messsystemen liefert die nominale Auflösung eines ADUs oft nicht die tatsächliche Genauigkeit. Die effektive Anzahl von Bits (ENOB) (engl. effective number of bits) quantifiziert die nutzbare Auflösung unter Berücksichtigung von Störungen, die durch interne (verschiedene Rauschquellen, Nichtlinearitäten, Apertur-Jitter) und externe Faktoren (dynamische Signalparameter, Umgebungsbedingungen) bedingt sind. Ein detailliertes Verständnis dieser Einflüsse ist unerlässlich, um die ADU-Leistung in der jeweiligen Anwendung zu optimieren.

Die Beziehung zwischen ENOB und SINAD

Der ENOB wird typischerweise aus dem gemessenen SINAD-Wert des ADUs abgeleitet. Die Formel zur Berechnung des ENOB aus dem SINAD lautet:

$$ENOB = \frac{SINAD_{dB} - 1,76 \text{ dB}}{6,02 \text{ dB/Bit}} \quad (2.10)$$



Beispiel 2.4

Ein Sensor erzeugt ein analoges Ausgangssignal, das linear im Bereich von 0 V bis 5 V variiert und eine bestimmte physikalische Größe repräsentiert. Dieses Signal wird mit einem ADU digitalisiert, dessen Eigenschaften wie folgt sind:

- nominale ADU-Auflösung: 16 Bit,
- gemessenes Signal-Rausch-und-Verzerrungs-Verhältnis (SINAD): 90 dB,
- voller Eingangsspannungsbereich: 5 V.

Basierend auf diesem SINAD beträgt der ENOB dieses ADUs unter den gegebenen Bedingungen:

$$ENOB = \frac{90 \text{ dB} - 1,76 \text{ dB}}{6,02 \frac{\text{dB}}{\text{Bit}}} \approx 14,66 \text{ Bit}$$

Die effektive Anzahl an Unterscheidungsstufen beträgt etwa 25.800, was einer minimalen detektierbaren Spannungsänderung von 194 μV entspricht. Im Gegensatz zur theoretischen Auflösung von 65 536 Stufen (16 Bit) begrenzt der ENOB die praktisch nutzbare Information auf das Niveau eines idealen 14,66-Bit-Wandlers.

Der ENOB ist entscheidend für die Bewertung der realen ADU-Leistung. Während ein hoher ENOB in High-End-Anwendungen wie Messtechnik, Kommunikation und Forschung unerlässlich ist, spielt er in robusteren Systemen oder bei vorverarbeiteten Signalen eine geringere Rolle. Die stetige Optimierung von ADUs durch Design, Kalibrierung und neue Technologien verbessert den ENOB und damit die Messgenauigkeit. Zukünftige Signalverarbeitungstechniken, wie Machine Learning zur Echtzeitkalibrierung und Rauschunterdrückung, versprechen weitere Fortschritte. Die Erforschung des ENOB bleibt somit ein dynamisches Feld für präzisere Messsysteme.

2.5 Codierungen für Datenwandler

AD- und DA-Wandler mit nur positiver Eingangs- und Ausgangsspannung (wie bisher betrachtet) werden als **unipolare Wandler** bezeichnet. Die Codierung der Zahl N am Ausgang bzw. Eingang ist üblicherweise binär und wird deshalb als „**binär**“ oder auch „**straight binär**“ bezeichnet. Es gibt aber durchaus auch andere Codierungen, z. B. mit BCD-Code (dies wird hier nicht weiter betrachtet).

AD- und DA-Wandler mit einem negativen und positiven Bereich für Eingangs- und Ausgangsspannung werden dann entsprechend als **bipolare Wandler** bezeichnet. Die bipolare Kennlinie ergibt sich, indem die Spannung durch einen Offset von $-\frac{1}{2}U_{\text{ref}}$ in den negativen Bereich verschoben wird. Die Achse für die Zahl N bleibt davon unberührt. Bezüglich der Codierung wird der Begriff „**offset-binär**“ verwendet.

In Tabelle 2.3 ist dies nochmal für einen 8-Bit-Wandler zusammengefasst.

Tabelle 2.3 Datenwandler-Codierungen

Datenwandler	Spannungsbereich	Codierung für N
binär	0 V bis $+FS$ (mit $FS = U_{\text{ref}}$)	0000 0000 bis 1111 1111
offset-binär	$-FS$ bis $+FS$ (mit $2 \cdot FS = U_{\text{ref}}$)	0000 0000 bis 1111 1111

Die Gleichungen für bipolare Wandler ergeben sich in Erweiterung der Formel 2.2 und Formel 2.1 zu:

Bipolarer DAU:

$$U = U_{\text{LSB}} \cdot N - \frac{1}{2} \cdot U_{\text{ref}} \quad (2.11)$$

Bipolarer ADU:

$$N = \left(\frac{U + \frac{1}{2} \cdot U_{\text{ref}}}{U_{\text{LSB}}} \right) \quad (2.12)$$

In Bild 2.11 und Bild 2.12 sind die Kennlinien für einen **bipolaren** 3-Bit-DAU mit $U_{\text{ref}} = 2,4 \text{ V}$ und einen bipolaren 3-Bit-ADU mit $U_{\text{ref}} = 4 \text{ V}$ dargestellt.

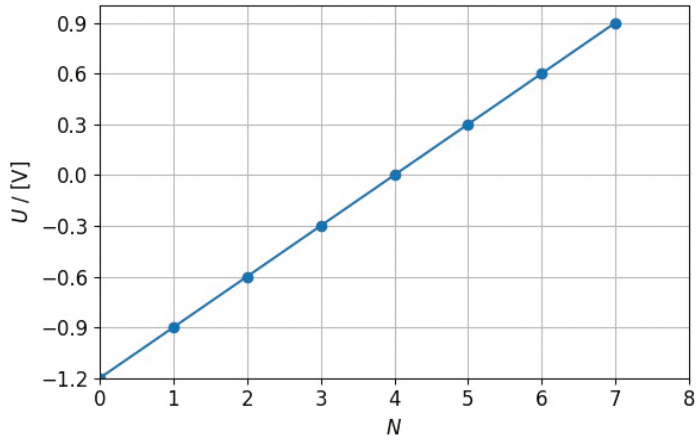


Bild 2.11 Bipolarer DAU

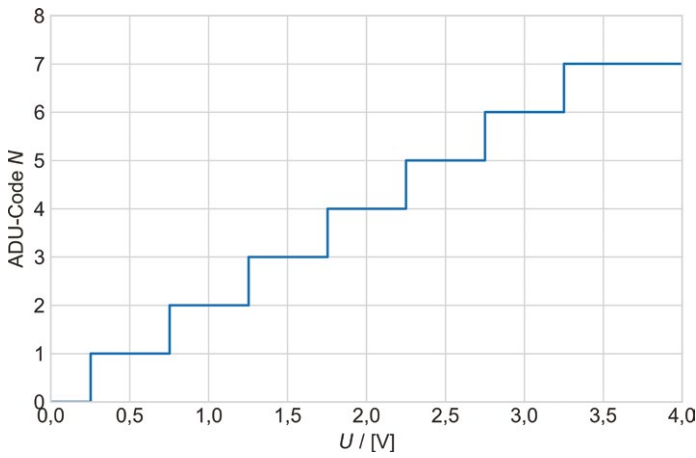


Bild 2.12 Bipolarer ADU

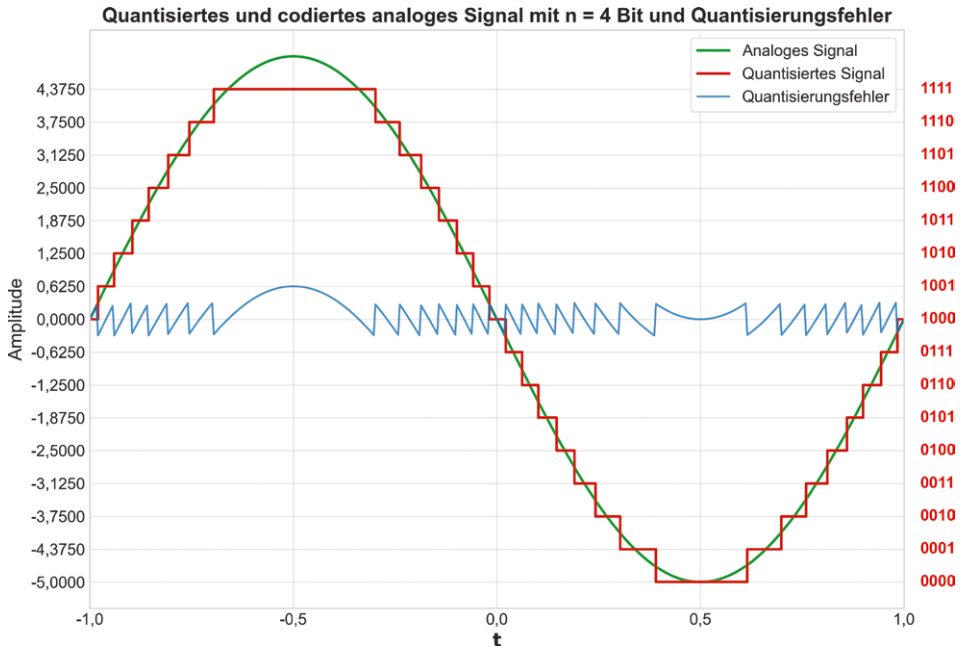


Bild 2.13 Quantisiertes analoges Signal mit $n=4$ Bit und Quantisierungsfehler



Beispiel 2.5

Der Arduino Uno besitzt einen 10-Bit-Analog-Digital-Wandler (ADC), der digitale Werte von 0 bis 1023 liefert und Spannungen im Bereich von 0 V bis 5 V messen kann. Wie kann eine bipolare Sinusspannung von $-2,5$ V bis $+2,5$ V mit dem Arduino digitalisiert werden? Bestimmen Sie den maximalen Wert des Sinussignals, das digitalisiert wird.

Lösung

Die Sinusspannung von $-2,5$ V bis $2,5$ V muss um $+2,5$ V verschoben werden, so dass sie in den Bereich von 0 V bis 5 V fällt.

Der Arduino Uno-ADU hat eine Auflösung von 10 Bit, was bedeutet, dass er Werte von 0 bis 1023 ausgibt (insgesamt 1024 Stufen).

$$\text{Maximaler Wert} = 5 \text{ V} \cdot \frac{1023}{1024} = 4,995 \text{ V}$$

Der maximale Wert für eine Sinusspannung von $2,5$ V nach der Verschiebung beträgt daher:

$$\text{Maximaler Wert} = 4,995 \text{ V} - 2,5 \text{ V} = 2,495 \text{ V}$$

2.6 AD-Umsetzung bei zeitveränderlicher Eingangsspannung

Jeder AD-Wandler benötigt die für ihn charakteristische Wandlungszeit t_c (engl. conversion time), um aus der anliegenden Spannung U den Zahlenwert N am Ausgang zu erzeugen. Ändert sich während der Wandlung die Eingangsspannung, so führt dies zu einem Fehler im ausgegebenen Wert N des ADU. N entspricht dann nicht dem Spannungswert zum Zeitpunkt der Wandlungsauslösung (Bild 2.14).

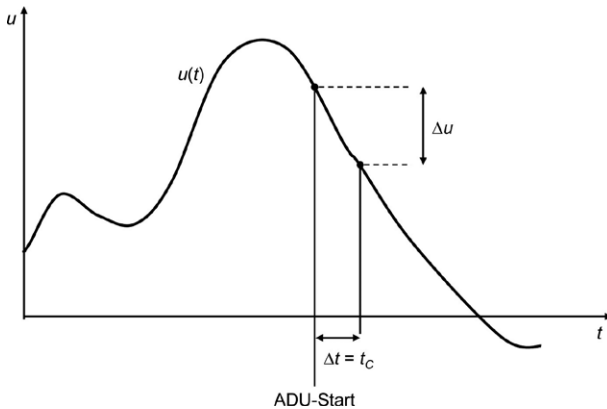


Bild 2.14
AD-Wandlung bei veränderlicher Eingangsspannung

Um den Fehler in der Ausgangsgröße auf ± 1 Bit zu beschränken, sollte während der Wandlungszeit t_c folgende Bedingung eingehalten werden:

$$\Delta u \leq \frac{1}{2} \cdot U_{\text{LSB}} \quad (2.13)$$

während der Wandlungszeit t_c

Die Formel 2.13 lässt sich anschaulicher formulieren, wenn ein sinusförmiges Eingangssignal betrachtet wird (Bild 2.15).

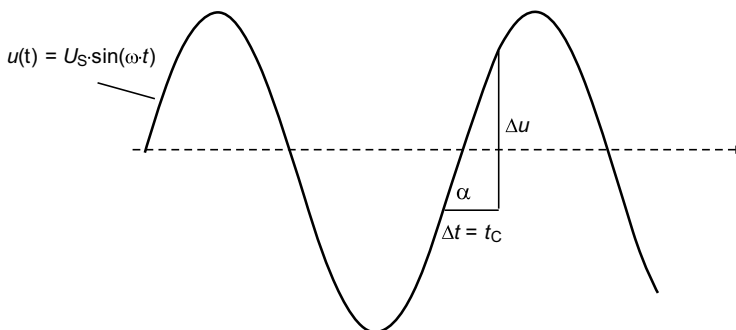


Bild 2.15 Sinusförmiges Eingangssignal

Für das in Bild 2.15 eingezeichnete Steigungsdreieck lässt sich folgender Ansatz machen:

$$\frac{\Delta u}{t_C} = \tan(\alpha) \approx \frac{du}{dt} = \frac{d(U_S \cdot \sin(\omega \cdot t))}{dt} = U_S \cdot \cos(\omega \cdot t) \cdot \omega$$

für $t = 0$ (Nulldurchgang der Sinusfunktion) gilt dann wegen $\cos(\omega t) = 1$:

$$\frac{\Delta u}{t_C} = U_S \cdot \omega$$

oder

$$\Delta u = t_C \cdot U_S \cdot \omega$$

Aus der Bedingung (Formel 2.13) folgt somit

$$t_C \cdot U_S \cdot \omega \leq \frac{1}{2} \cdot U_{\text{LSB}}$$

und mit $\omega = 2 \cdot \pi \cdot f$ schließlich die Forderung an die Frequenz eines sinusförmigen Eingangssignals:

$$f \leq \frac{U_{\text{LSB}}}{4 \cdot \pi \cdot U_S \cdot t_C} \quad (2.14)$$



Beispiel 2.6

Welche maximale Frequenz ist bei einem 8-Bit-ADU (bipolar; $FS = 5 \text{ V}$; $t_C = 5 \mu\text{s}$) für Vollaussteuerung mit Sinussignal zulässig?

Lösung

$$U_{\text{LSB}} = \frac{2 \cdot FS}{2^n} = \frac{10 \text{ V}}{2^8} = 39,06 \text{ mV}$$

$$f_{\text{max}} = \frac{U_{\text{LSB}}}{4 \cdot \pi \cdot U_S \cdot t_C} = \frac{U_{\text{LSB}}}{4 \cdot \pi \cdot 5 \text{ V} \cdot 5 \mu\text{s}} = 124 \text{ Hz}$$

Für Eingangssignale mit höheren Frequenzen ist dem AD-Umsetzer ein SH-Schaltkreis (Sample-Hold-Schaltkreis = Abtast-und-Halte-Schaltkreis) vorzuschalten.

2.7 Sample-Hold-Schaltung

Der Sample-Hold-Schaltkreis (Bild 2.16) ist ein Analogwert-Speicher, der den Momentanwert einer Spannung für einen nachfolgenden kurzen Zeitraum in einer Kapazität speichert. Eingesetzt wird der SH-Schaltkreis immer dann, wenn sich die Eingangsspannung eines ADU während der Umsetzungszeit zu stark (siehe Formel 2.13) ändert. Die SH-Schaltung hat dann die Aufgabe, zu einem bestimmten Zeitpunkt – dem