

# Embedded Systems

## Kapitel 6: Analoge Eingabe

**Prof. Dr. Wolfgang Mühlbauer**

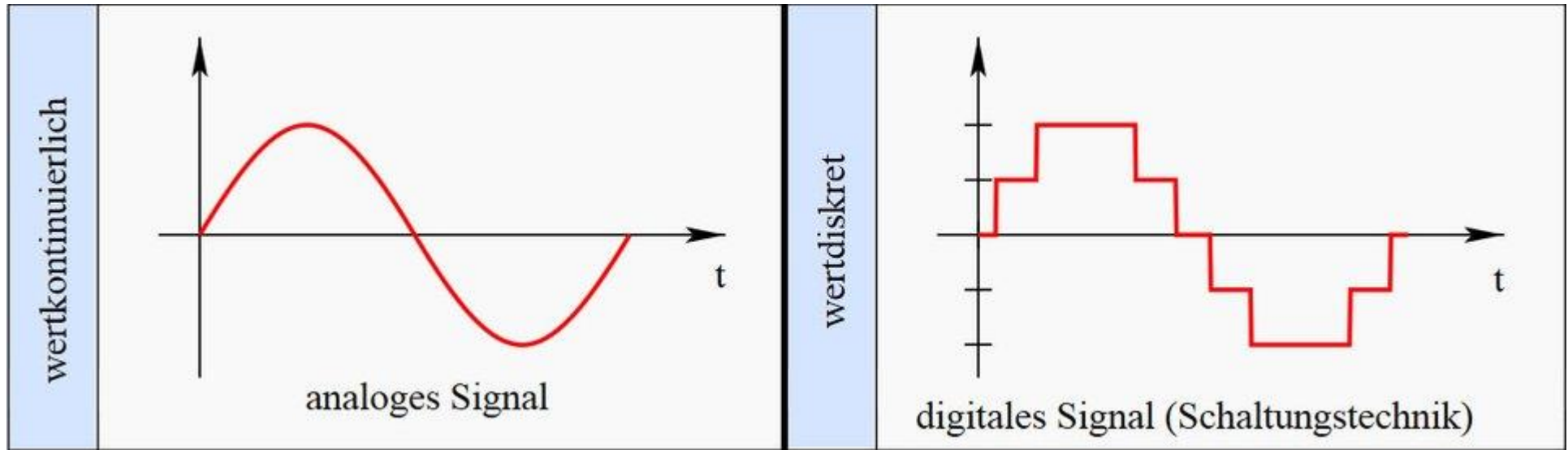
Fakultät für Informatik

`wolfgang.muehlbauer@th-rosenheim.de`

**Sommersemester 2020**

- ❑ **Einführung**
- ❑ Arbeitsweise von A/D Wandlern
- ❑ A/D Umsetzung beim ATmega2560

# Analog vs. Digital



Quelle: <https://www.giga.de/extra/ratgeber/specials/unterschied-zwischen-analog-und-digital-einfach-erklart/>

## ❑ **Abtastung / Zeitliche Diskretisierung**

- Werte Signal nur nach Ablauf von (festen) Zeitintervallen aus.

## ❑ **Quantisierung**

- Messe dabei die Werte nur mit einer bestimmten Genauigkeit.

# Analoge Ein- und Ausgabe: Beispiele

## ❑ **Analoge Eingabe:** Sehr oft trägt ein analoger Spannungswert Information

- Druckempfindliche Buttons (IBM Trackpoint)
- Potentiometer
- DMS-Streifen
- Temperatursensor
- Fotodiode
- ...

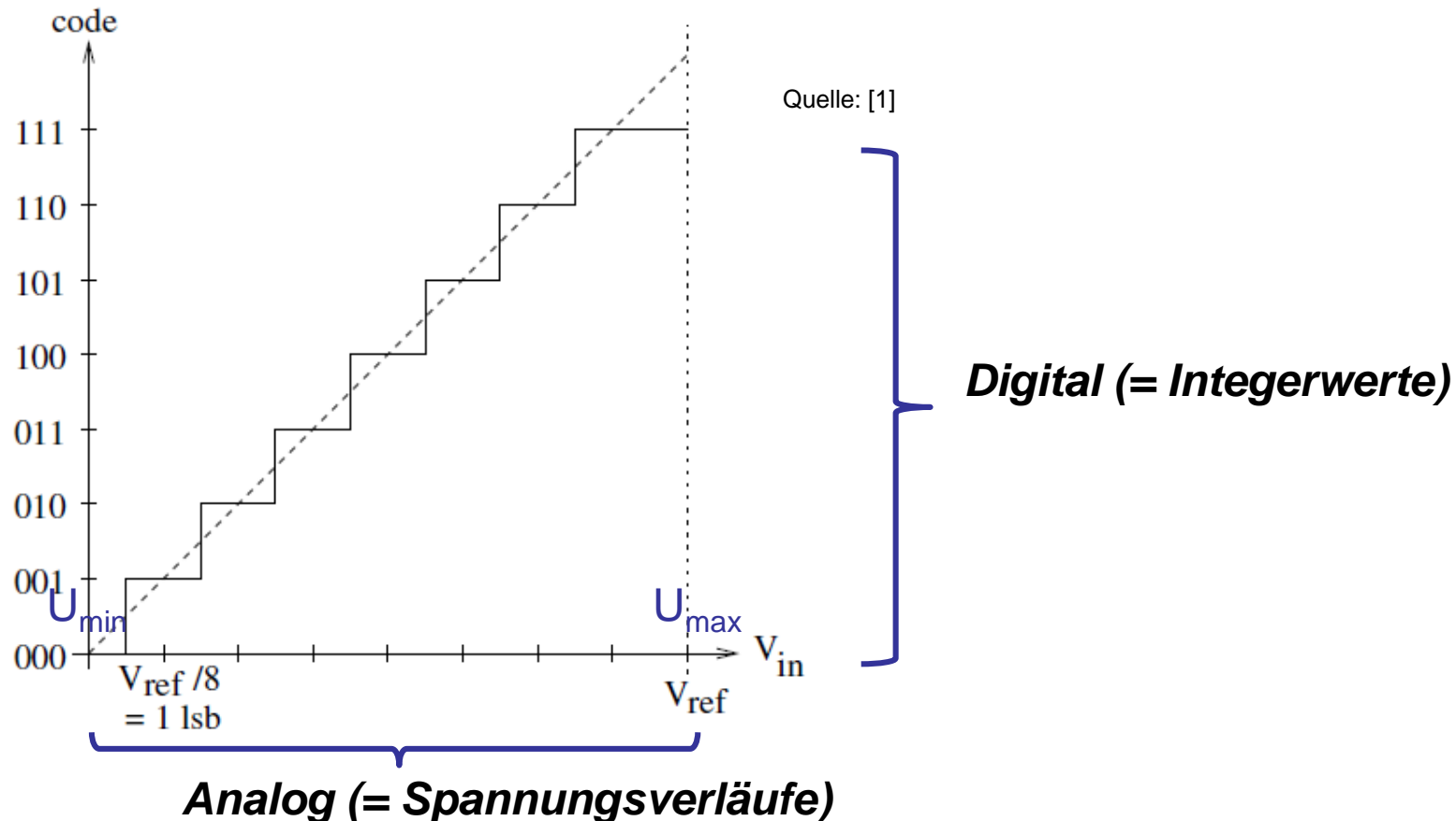
## ❑ **Analoge Ausgabe**

- Beispiele: Helligkeitskontrolle einer LED, Ansteuern eines Motors
- In der Praxis oft: Pulsweitenmodulation → siehe letztes Kapitel.

# A/D Umsetzung

## □ Abbildung von

- *analogem Wertebereich*  $[U_{\min}; U_{\max}]$  (meist Spannungsbereich) auf
- *binärem Digitalwert (Code)* der Länge  $r$  (Dualzahl)



# A/D bzw. D/A Umsetzer

## □ **A/D Wandler** (auch A/D Umsetzer)

- Muss Spannung  $V$  in digitalen Wert  $B = (b_{r-1}..b_0)$ ,  $r \geq 1$  umsetzen.
- Sicht des Mikrocontrollers: Eingaberichtung
- Implementierung über D/A Umsetzer (Rückkopplung)

## □ **D/A Wandler** (auch D/A Umsetzer)

- Muss digitalen Wert  $B = (b_{r-1}..b_0)$ ,  $r \geq 1$  in eine proportionale Spannung  $V$  umwandeln.
- Sicht des Mikrocontrollers: Ausgaberichtung
- Widerstandsnetzwerk

- ❑ Einführung
- ❑ **Arbeitsweise von A/D Wandlern**
- ❑ A/D Umsetzung beim ATmega2560

## □ Ohmsches Gesetz

- $R = U / I$

## □ Schaltung

- Widerstände  $R_1$  und  $R_2$  sind parallel.
- Strom  $I$  teilt sich auf.

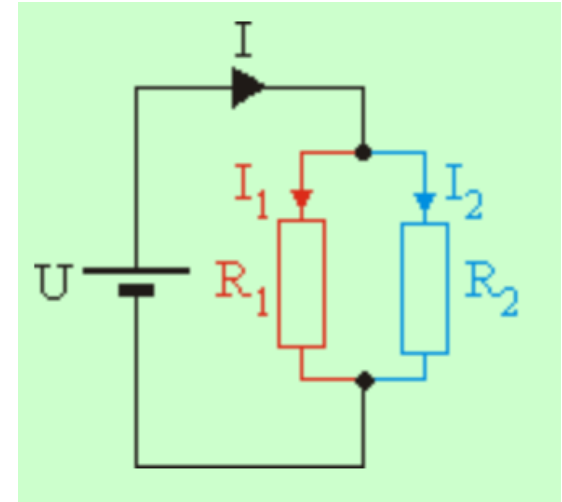
## □ Frage:

- Wie groß Ersatzwiderstand  $R_{1,2}$ , damit insgesamt gleicher Strom?
- Herleitung durch Ohmsches Gesetz

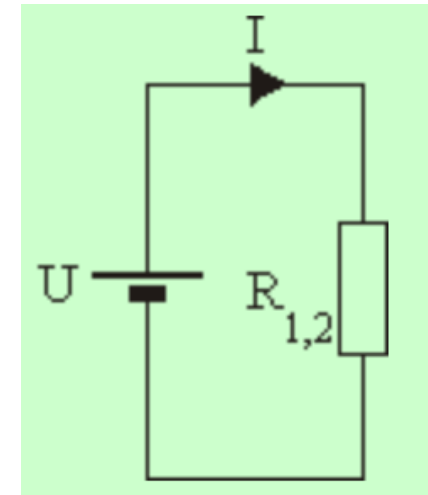
## □ Feststellung

- $R_{1,2} = \frac{R_1 \cdot R_2}{R_1 + R_2}$
- Falls  $R_1 = R_2 \rightarrow R_{1,2} = \frac{R_1}{2}$

## Parallelschaltung



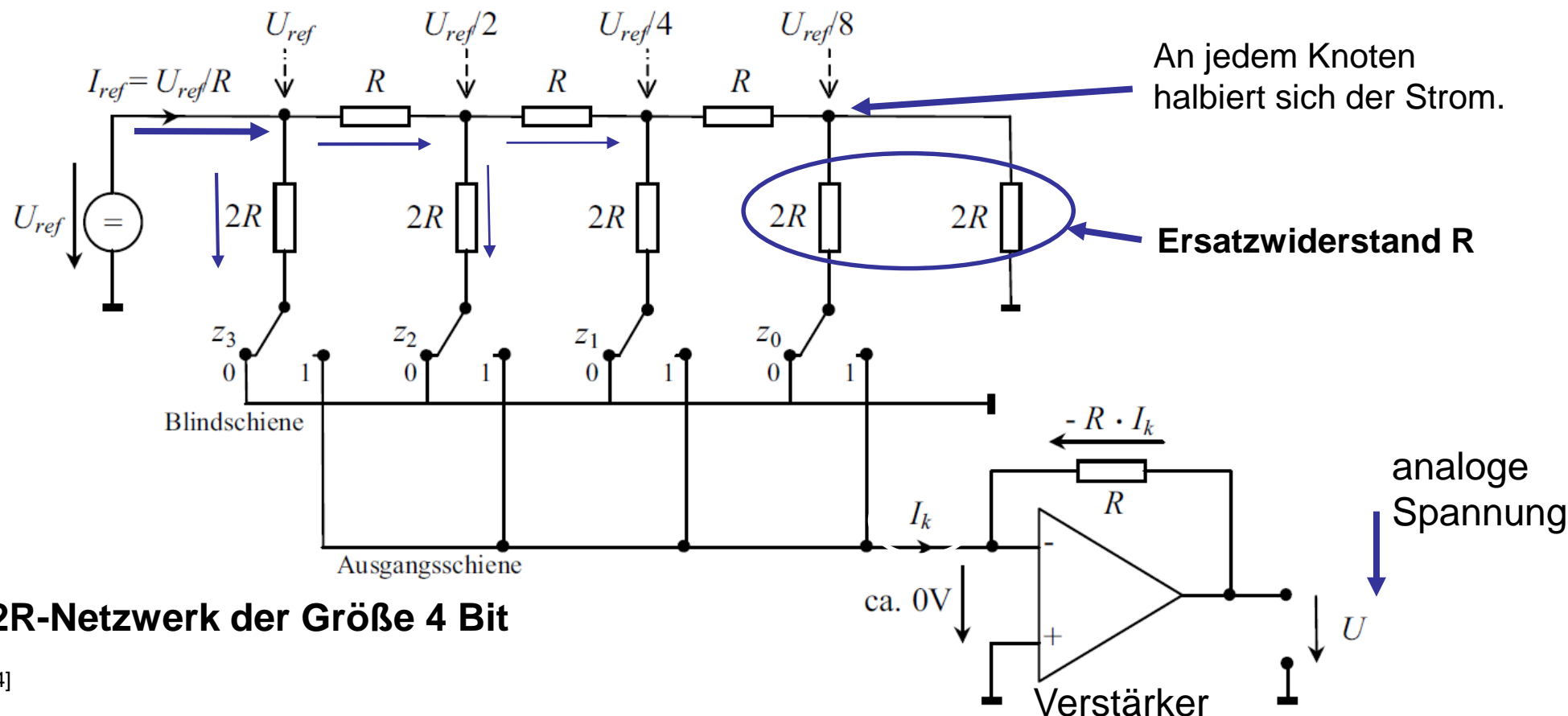
## Ersatzwiderstand



Quelle [5]



- ❑ **Widerstandsnetzwerk:** Widerstände der Größe  $R$  und  $2R$ .
- ❑ Es werden Teilspannungen der Referenzspannung  $U_{ref}$  im Verhältnis der Wertigkeit der Binärziffern  $z_0$  bis  $z_3$  der Dualzahl ( $z_3 z_2 z_1 z_0$ ) erzeugt.



Quelle: [4]

# A/D Umsetzung

## □ Lineare Abbildung

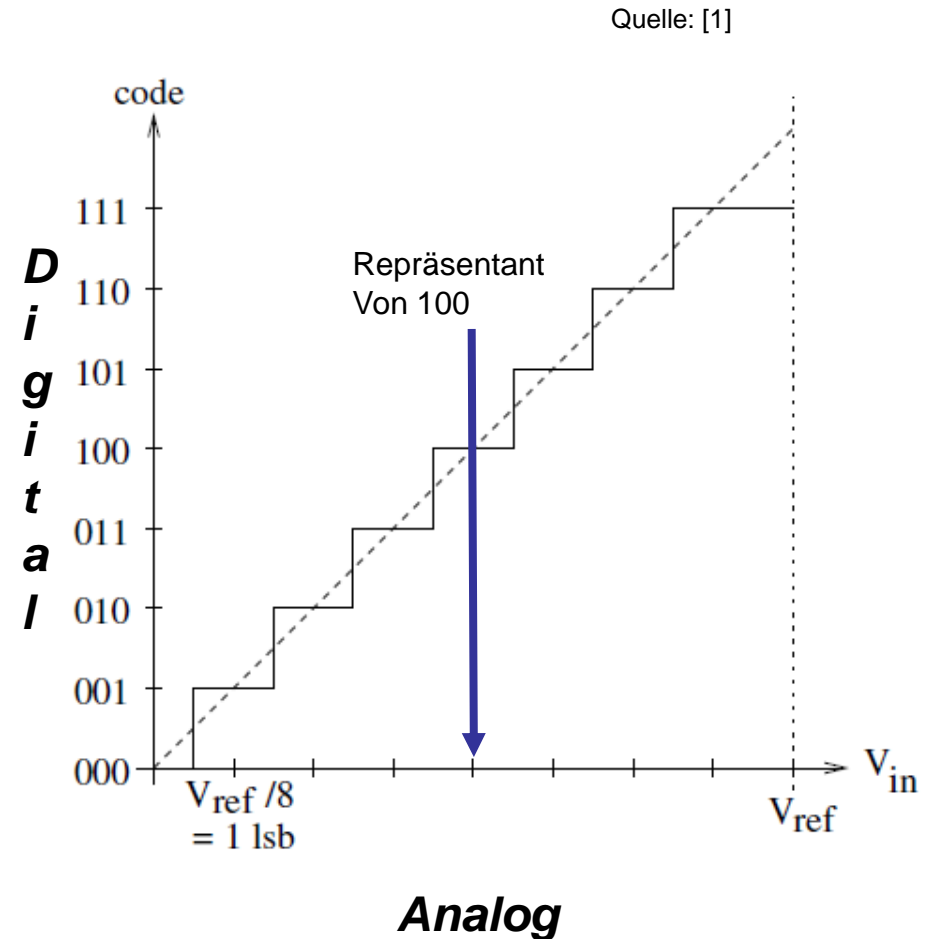
- Analoge Eingangsspannung  $V_{in}$  aus dem Bereich  $[U_{min}; U_{max}]$ , meist  $[0; V_{ref}]$
- auf binären Code mit  $r$  Bits

## □ Auflösung eines A/D Wandlers

- Länge des binären Codes
  - Hier:  $r = 3$
- Kleinster Spannungsunterschied, *Least Significant Bit* (LSB)
  - Hier:  $LSB = V_{ref} / 2^r$

## □ Um Fehler zu minimieren liegt "Repräsentant" eines Intervalls immer in Intervallmitte; Ausnahme:

- 1. Intervall: Repräsentant des Codes 000 sollte 0V sein → Stufenbreite  $\frac{1}{2}$  LSB.
- Letztes Intervall: Stufenbreite  $1\frac{1}{2}$  LSB



- ❑ Handbuch: Seite 268
- ❑ Welche **Auflösung** hat der ATmega2560 Mikroprozessor, falls die Referenzspannung 5V beträgt?
- ❑ Ein 8-Bit A/D Wandler wird mit der Referenzspannung 5V betrieben. Die Eingangsspannung sei 3,5V. Wie lautet der resultierende binäre Code?

# Exkurs: Fehlerquellen bei A/D Umsetzung

## Quantisierungsrauschen (1)

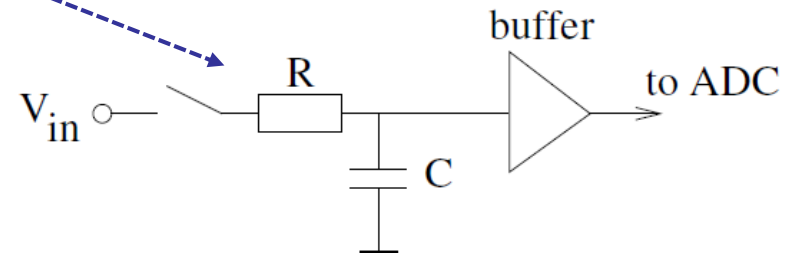
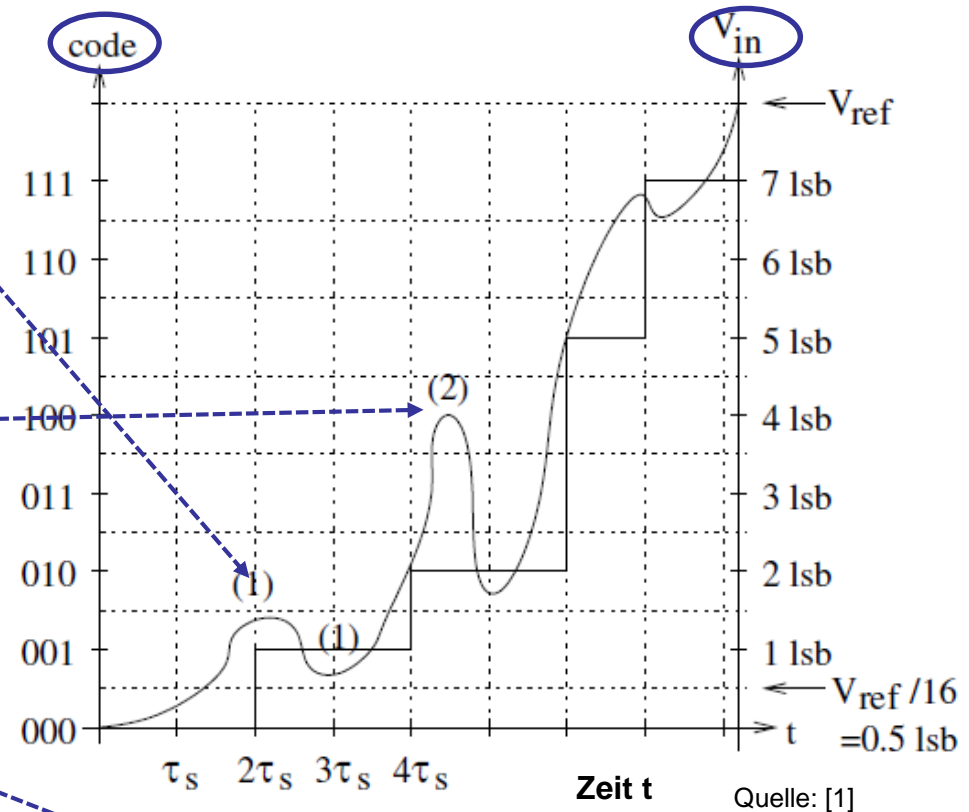
- Abweichung des gewandelten treppenförmigen vom tatsächlichen stufenlosen Analogsignals.

## Umsetzungszeit (2)

- Umsetzung benötigt Zeit, manche Änderungen werden übersehen.

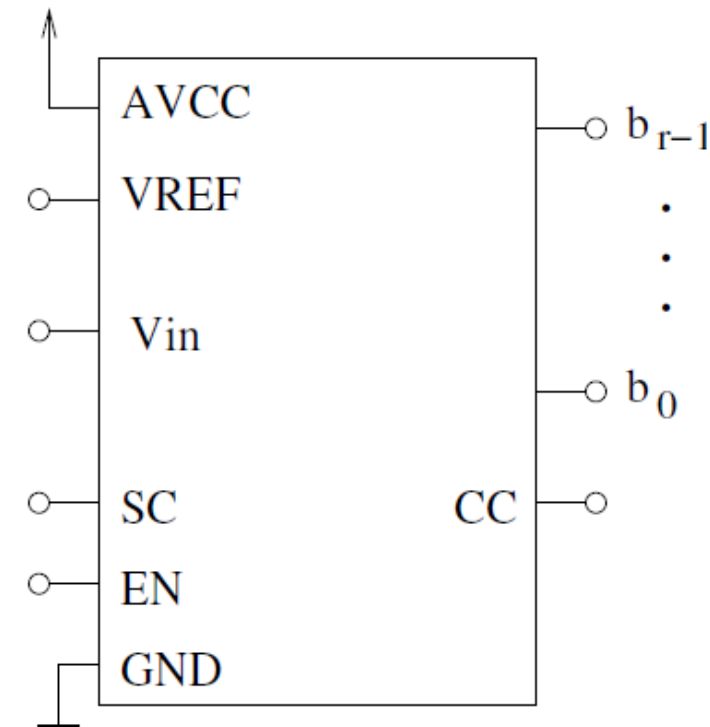
## Änderungen des Eingangs während der Umsetzung

- Abhilfe: **Abtast/Halteglieder** halten analoge Spannung während der Umsetzung konstant.



# A/D Wandler als Blackbox: Integrated Circuit (IC)

- ❑ AVCC und GND
  - Stromversorgung des A/D Wandlers
- ❑ VREF
  - Maximale, analoge Spannung
  - Definiert den analogen Wertebereich von  $[0; V_{REF}]$
- ❑  $V_{in}$ 
  - Eingang
  - Analoger Spannungswert, der umgesetzt werden soll.
- ❑ EN
  - Schaltet den Baustein ein.
- ❑ SC
  - Startet eine Umsetzung



Quelle: [1]

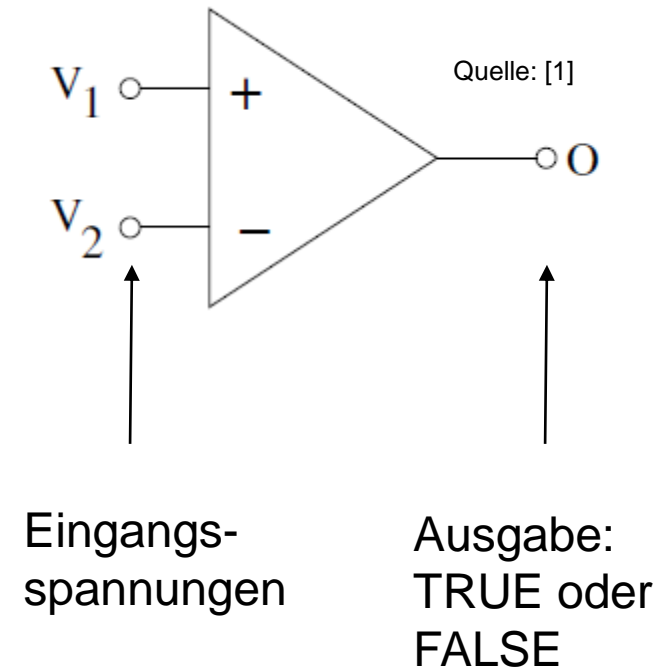
# Wie implementiert man A/D Umsetzer in Hardware?

---

- ❑ Komparator (1 Bit)
- ❑ Parallelverfahren
- ❑ Zählverfahren
- ❑ Wägeverfahren

# Komparator

- ❑ Bauelement, das 2 Eingangsspannungen miteinander vergleicht.
  - $V_1 > V_2 \rightarrow$  Ausgabe: HIGH
  - $V_1 < V_2 \rightarrow$  Ausgabe: LOW
- ❑ Unterschied zu digitalem Eingangspin
  - Vergleicht **2** Spannungen
  - Kein Schmitt-Trigger
- ❑ In den meisten Mikrocontroller vorhanden
  - ATmega2560: Kapitel 25
  - $AIN0 > AIN1 \rightarrow AC0$  ist HIGH
- ❑ Komparator = **1-Bit A/D Wandler**
  - Grundlage für alle weiteren Verfahren



# Parallelverfahren

## ❑ Idee

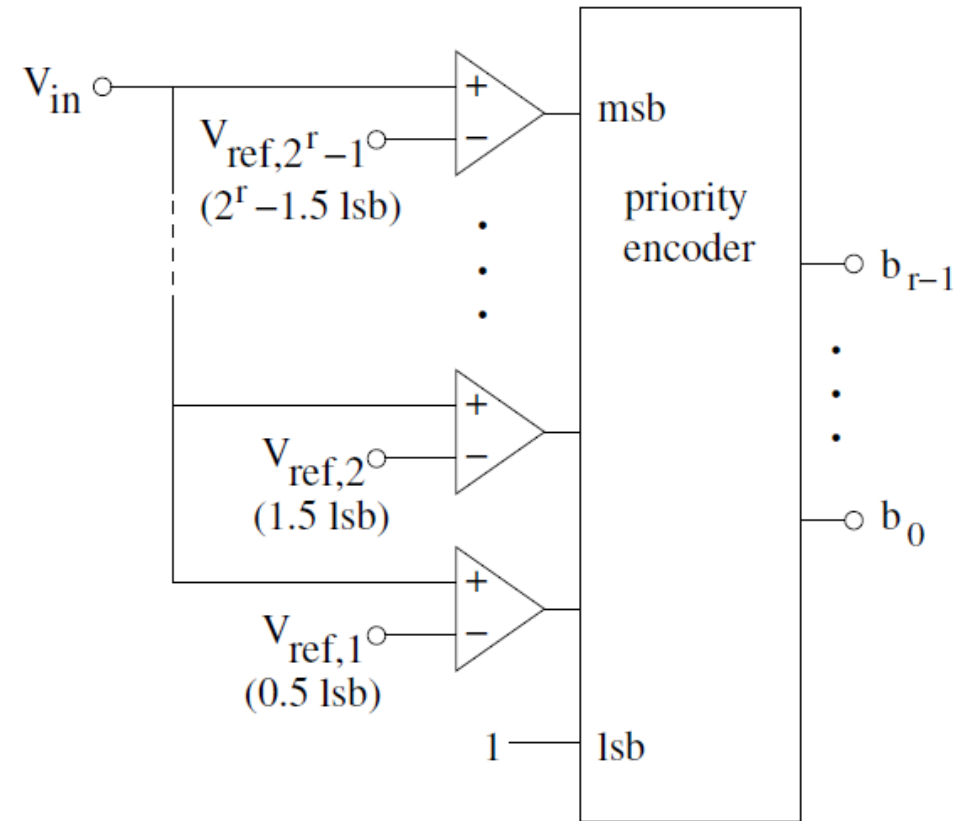
- $2^r - 1$  Komparatoren, 1 Komparator für jede "Flanke" bzw. Stufengrenze
- Vergleiche gegen  $2^r - 1$  Referenzspannungen
- Ergebnis der Vergleiche ist Input für Encoder.
- Encoder gibt die Binärzahl ( $b_0, \dots, b_{r-1}$ ) aus.

## ❑ Vorteil

- Umsetzung schnell, da parallel.

## ❑ Nachteil

- Aufwendig, da  $2^r - 1$  Komparatoren notwendig.



Quelle: [1]



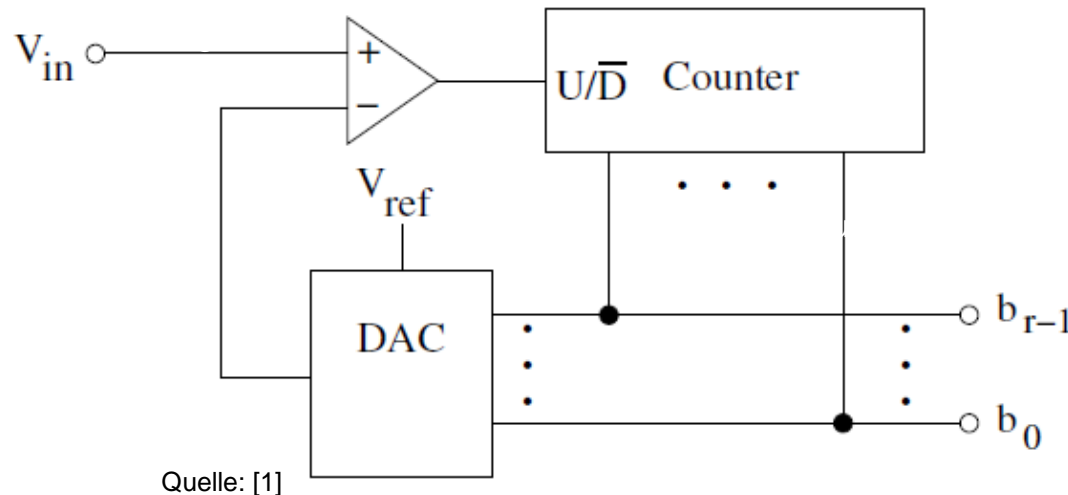
# Zählverfahren

## ❑ Idee

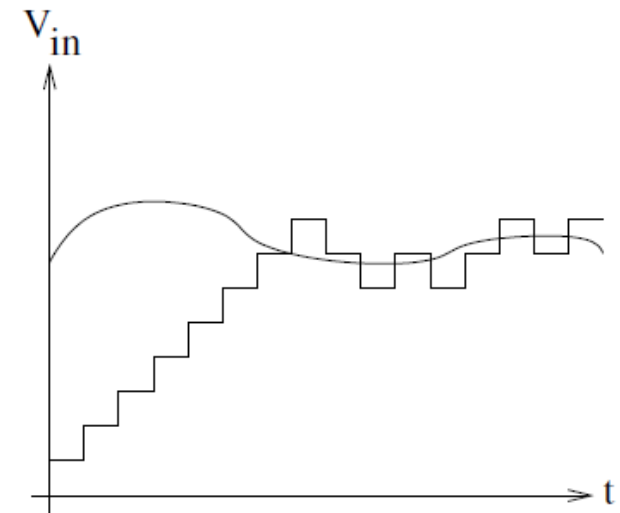
- **Verwende D/A-Wandler (DAC) für eine A/D Umsetzung!**
- *Digitaler Zähler* (Counter) speichert „Schätzung“ des digitalen Wertes
- DAC wandelt Schätzwert in analoge Spannung um
- Rückkopplung: Komparator vergleicht gegen Eingangsspannung und anschließendes Erhöhen bzw. Erniedrigen des Zählers.

## ❑ **Vorteil:** Einfache Realisierung

## ❑ **Nachteil:** Geschwindigkeit, lange Einschwingzeit.



**Zählverfahren: Blockschaltbild**



**Zählverfahren: Einschwingvorgang**

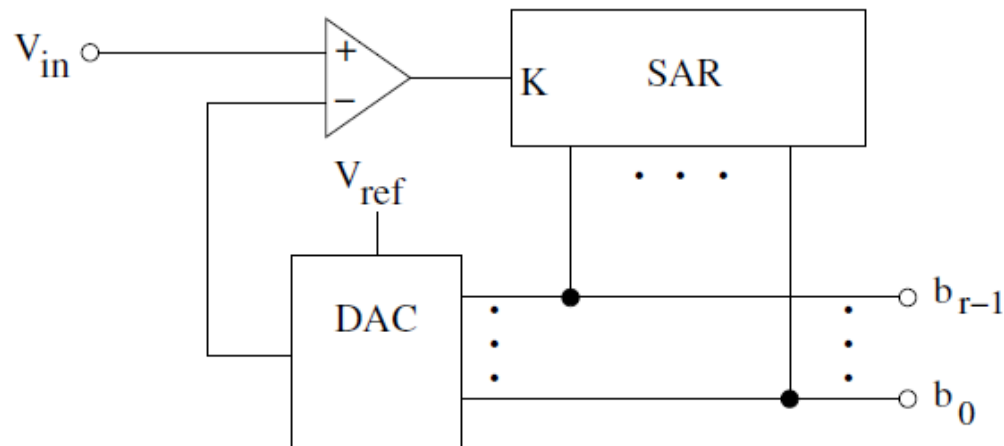
# Wägeverfahren

## ❑ Modifiziertes Zählverfahren

- Ersetze Zähler gegen **Successive Approximation Register (SAR)**
- SAR implementiert **binäre Suche** (in Hardware) anstatt einfach hochzuzählen.

## ❑ **Sukzessive Approximation** des Wertes der Eingangsspannung

- Vom MSB zum LSB (least significant Bit)
- Runde  $i$ : Setze Bitwert  $b_i$  auf 1, erzeugte Spannung größer als zu messende Spannung?
  - Ja: Behalte  $b_i=1$
  - Nein: Setze  $b_i=0$



Quelle: Gridling et al.

## ❑ **Beispiel:** $r=3$ ; $V_{in} \approx (1011)V$

- 1. Runde:  $b=1000$
- 2. Runde:  $b=1100$
- 3. Runde:  $b=1010$
- 4. Runde:  $b=1011$

- ❑ Einführung
- ❑ Arbeitsweise von A/D Wandlern
- ❑ **A/D Umsetzung beim ATmega2560**

# A/D Umsetzung beim ATmega2560 (1)

- ❑ 1 echter A/D-Umsetzer, mehrere analoge Eingangspins!
  - 16 Eingangspins können an A/D weitergeleitet werden.
  - Konfigurierbar, welcher Eingang an A/D weitergeleitet wird.
  
- ❑ **Trigger** einer A/D Umsetzung
  - **Manuell** durch Codeanweisung.
  - **Auto Trigger:** Hardware löst automatisch eine A/D Umsetzung aus
    - *Free Running Mode:* Sobald fertig, nächste Umsetzung → Endlosschleife.
    - Nach Timer Overflow Interrupts
    - Nach Änderungen am Komparatorausgang
  
- ❑ Wie erkennt man, dass A/D Umsetzung **beendet**?
  - Auswerten eines speziellen Flags, gesetzt durch ATmega
  - Durch speziellen Interrupt.
  - [Geänderter Wert steht in Datenregister]

# A/D Umsetzung beim ATmega2560 (2)

## ❑ **Prescaler**

- Interner A/D Umsetzer, der *spezielle Taktrate* benötigt.
- Konfigurierbar.

## ❑ **Referenzspannung**, Alternativen:

- Versorgungsspannung AVCC des Mikrocontrollers als Referenz (hier 5,0V)
- Feste interne Spannungen: 1,1V und 2,56 V
- Pin (AREF), an dem beliebige Spannung angelegt werden kann.

## ❑ Wertebereich der umsetzbaren Spannungswerte

- **Single-Ended Conversion:** Nur positive Spannungen  $[0, V_{ref}]$ 
  - Vergleiche Spannung an Eingangspin mit GND.
- **Differential Conversion:** Vorzeichenbehaftet  $[-V_{ref}/2, V_{ref}/2]$ 
  - Vergleiche Spannungen zwischen 2 Eingangspins.
  - Komplexer: Unterscheidung zwischen negativen und positiven Spannungswerten

# Wichtige Register für A/D Umsetzer

## ❑ **ADMUX**

- Wahl der Referenzspannung
- Wahl der analogen Eingangspins für A/D Umsetzung

## ❑ **ADCSRB**

- Wahl der analogen Eingangspins für A/D Umsetzung
- Single-Ended oder Differential Conversion
- Free Running Mode oder manuelles Triggern

## ❑ **ADCSRA**

- Aktivieren und Starten der A/D Umsetzung
- Prescaler
- Interrupts

## ❑ **ADCL** und **ADCH**

- Speichert Ergebnis der A/D Umsetzung
- Erst ADCL, dann ADCH lesen (atomarer Zugriff)

### **Achtung:**

Teilweise verteilen sich die logisch zusammengehörige Bits / Flags über mehrere Register.

# Quellenverzeichnis

- [1] G. Gridling und B. Weiss. *Introduction to Microcontrollers*, Version 1.4, 26. Februar 2007, Kapitel 2.5, verfügbar online:  
<https://ti.tuwien.ac.at/ecs/teaching/courses/mclu/theory-material/Microcontroller.pdf>  
(abgerufen am 08.03.2017)
- [2] Datenblatt ATmega2560, [http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf), (abgerufen am 19.03.2017)
- [3] M. Jimenez, R. Palomera und I. Couvertier. *Introduction to Embedded Systems*, Springer Verlag, 2014
- [4] U. Brinkschulte, T. Ungerer. *Mikrocontroller und Mikroprozessoren*, 3. Auflage, Springer Verlag, 2010
- [5] <http://www.leifiphysik.de/elektrizitaetslehre/komplexere-schaltkreise/ersatzwiderstand-bei-serien-und-parallelschaltung> (abgerufen am 01.05.2017)