Fachprojekt for Embedded System: Design and Implement Your Own Embedded Systems

Junjie Shi Niklas Ueter

LS 12, TU Dortmund

17,April,2019

Inhalt

Beispiel: System mit Sensor und LCD

1. Bibliothek importieren
2. Temperatur und Luftfeuchtigkeit auf einem LCD anzeigen
3. Übung

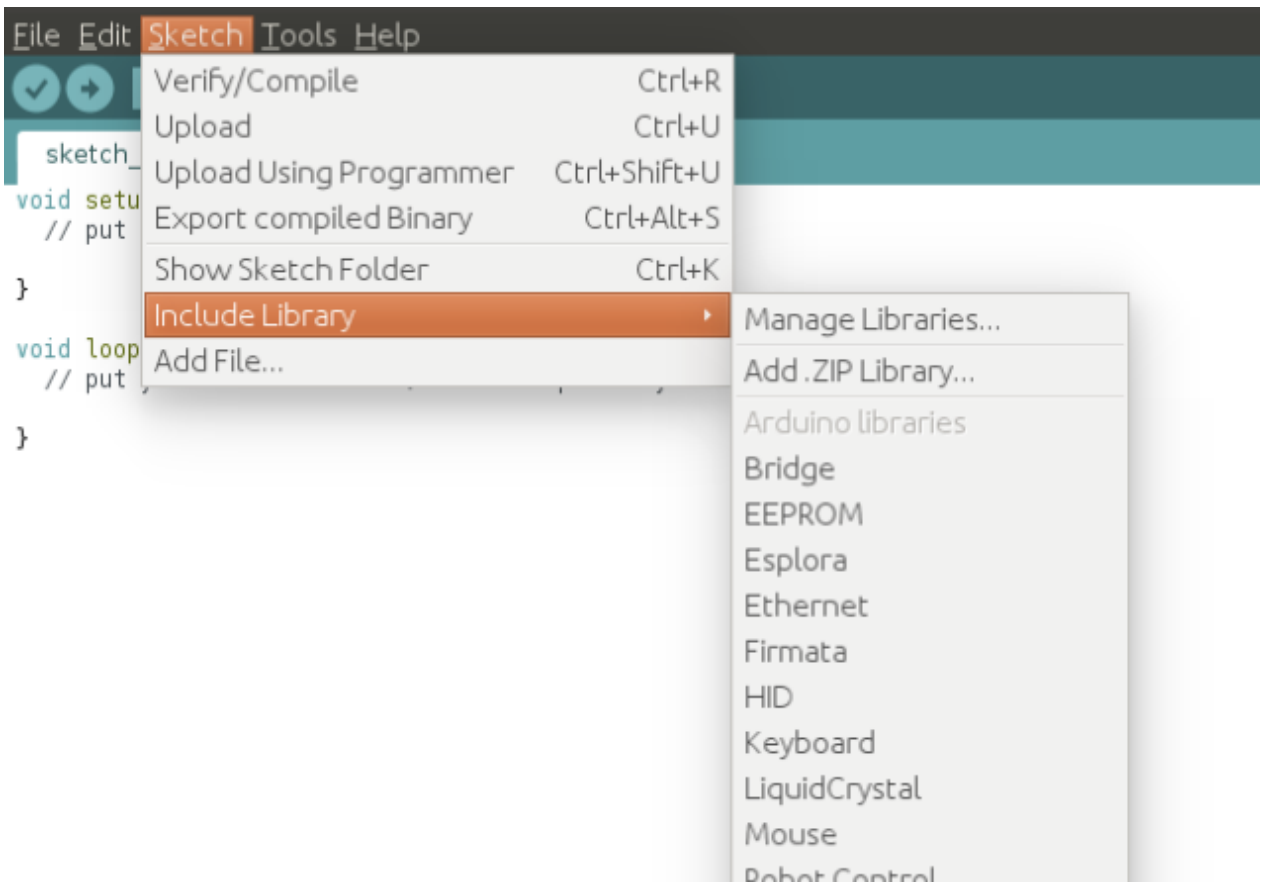
Weitere Referenzen

System mit Sensor und LCD: Voraussetzungen

- Zusätzliche Komponenten hinzufügen
- Komponentenspezifische Bibliotheken importieren

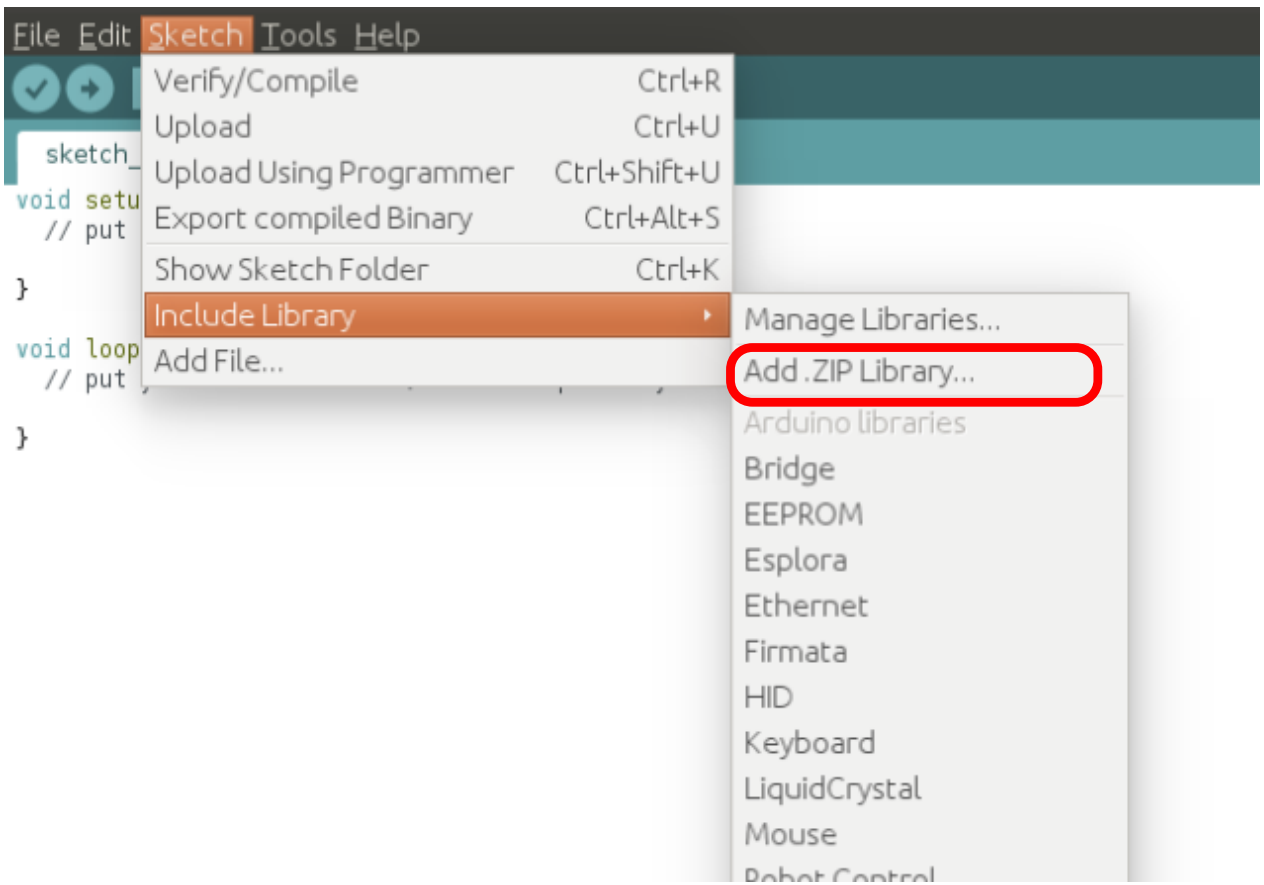
System mit Sensor und LCD: Voraussetzungen

- Zusätzliche Komponenten hinzufügen
- Komponentenspezifische Bibliotheken importieren



System mit Sensor und LCD: Voraussetzungen

- Zusätzliche Komponenten hinzufügen
- Komponentenspezifische Bibliotheken importieren



System mit Sensor und LCD

- Temperatur und Feuchtigkeitssensor DHT11
- LCD 1602 Modul
- Externe Bibliotheken werden benötigt

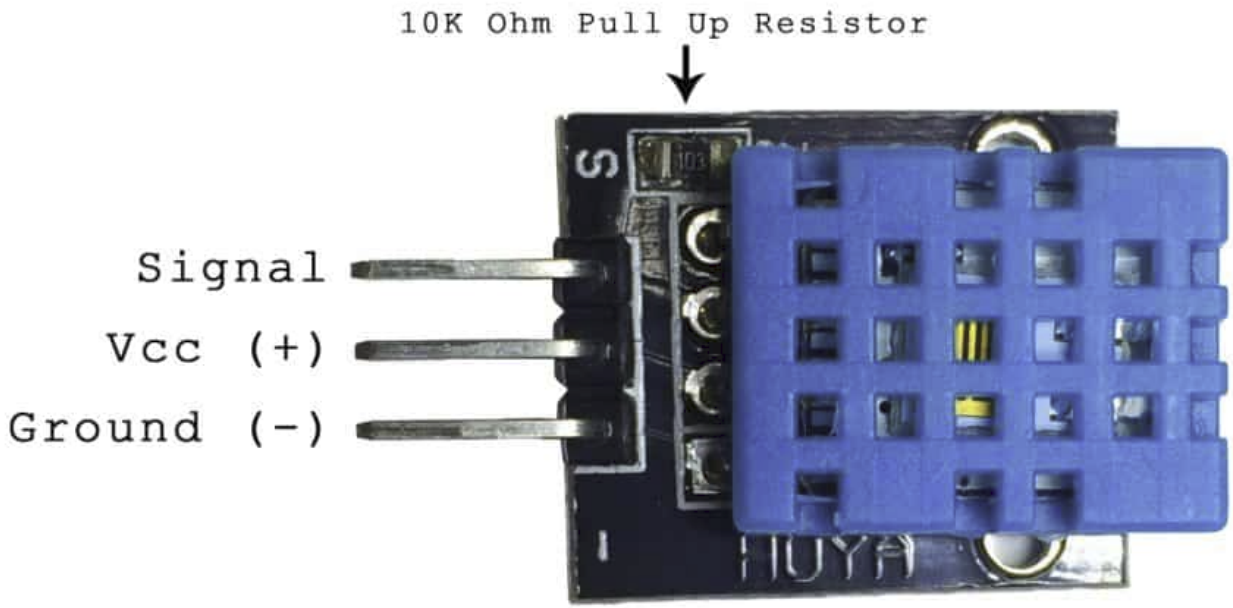
$\Rightarrow$ Download DHT11 Lib:

`http://www.circuitbasics.com/wp-content/uploads/2015/10/DHTLib.zip`

- Die Bibliothek für das LCD ist unter dem Namen *LiquidCrystal* bereits vorhanden

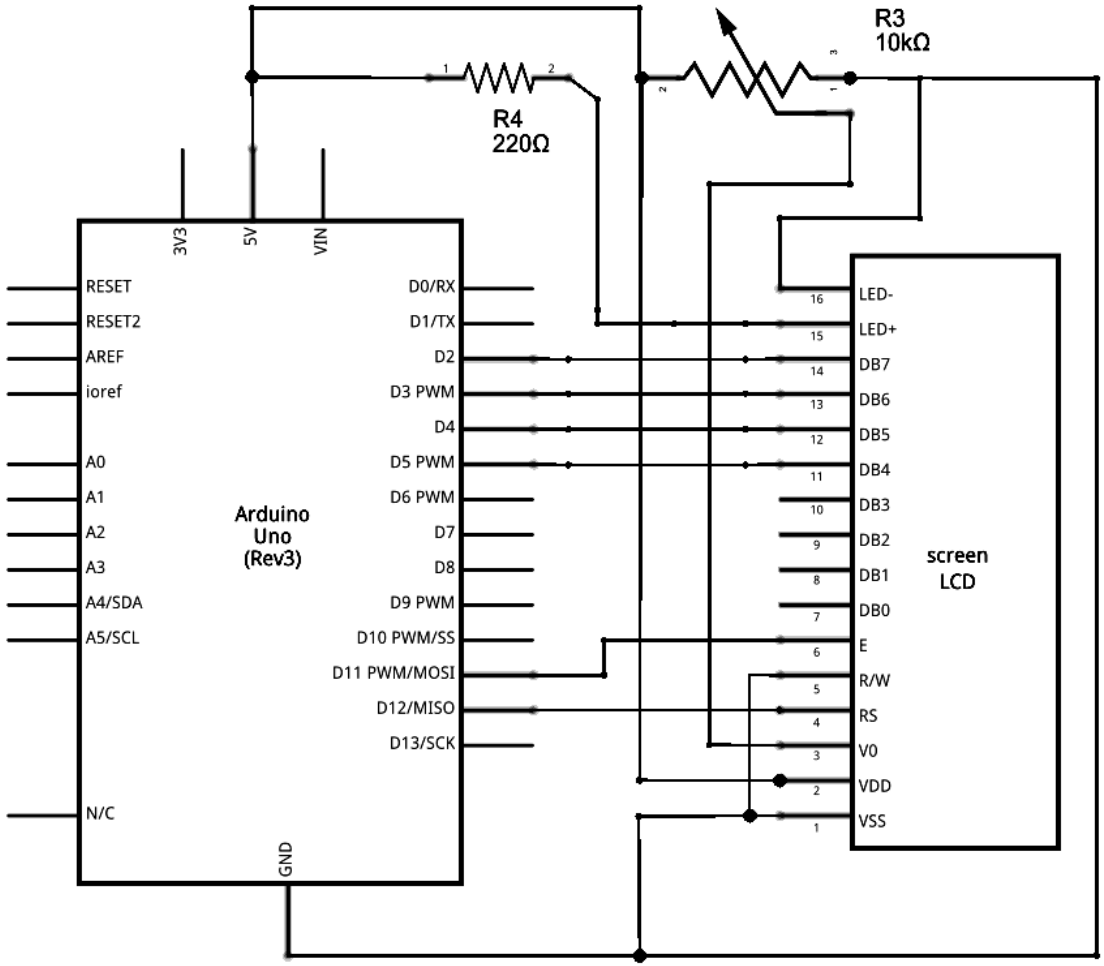
System mit Sensor und LCD

- Anschlüsse des DHT11 Sensors:



<ref: http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>

System mit Sensor und LCD



Ref: https://www.arduino.cc/en/Tutorial/HelloWorld

- **VSS**: Ground
- **VDD**: +5V Versorgungsspannung
- **V0**: Kontrasteinstellung des Displays
- **RS (Register Select)** wählt zwischen:
  1. Data Register: Daten, die auf den Bildschirm geschrieben werden (RS = 1).
  2. Instruction Register: Instruktion, die als nächstes ausgeführt werden soll (RS = 0).
- **R/W**: Auswahl zwischen Lesendem und Schreibendem Modus
- **Enable**: Ermöglicht das Beschreiben von Registern und wird verwendet um anzuzeigen, dass neue Daten (data register) vorhanden sind.
- **Daten**: $D_0, D_1, \ldots, D_7$ für Daten.

https://www.sunfounder.com/lcd1602-module.html>

Beispielcode I

```
#include <dht.h>
#include <LiquidCrystal.h>
#define DHT11_PIN 7

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
dht DHT;

void setup() {
 // set up the number of columns and rows
 lcd.begin(16, 2);
}

void loop() {
 //read the data from the senor
```

Beispielcode II

```
int chk = DHT.read11(DHT11_PIN);
//define the position of the screen in first line
lcd.setCursor(0,0);
lcd.print("Temp: ");
//print the temperature
lcd.print(DHT.temperature);
//print the symbol
lcd.print((char)223);
lcd.print("C");
//define screen position in second line
lcd.setCursor(0,1);
lcd.print("Humidity: ");
lcd.print(DHT.humidity);
lcd.print("%");
//delay for refresh
```

Beispielcode III

```
  delay(2000);
}
```

<ref: http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>

System mit Sensor und LCD: Demonstration

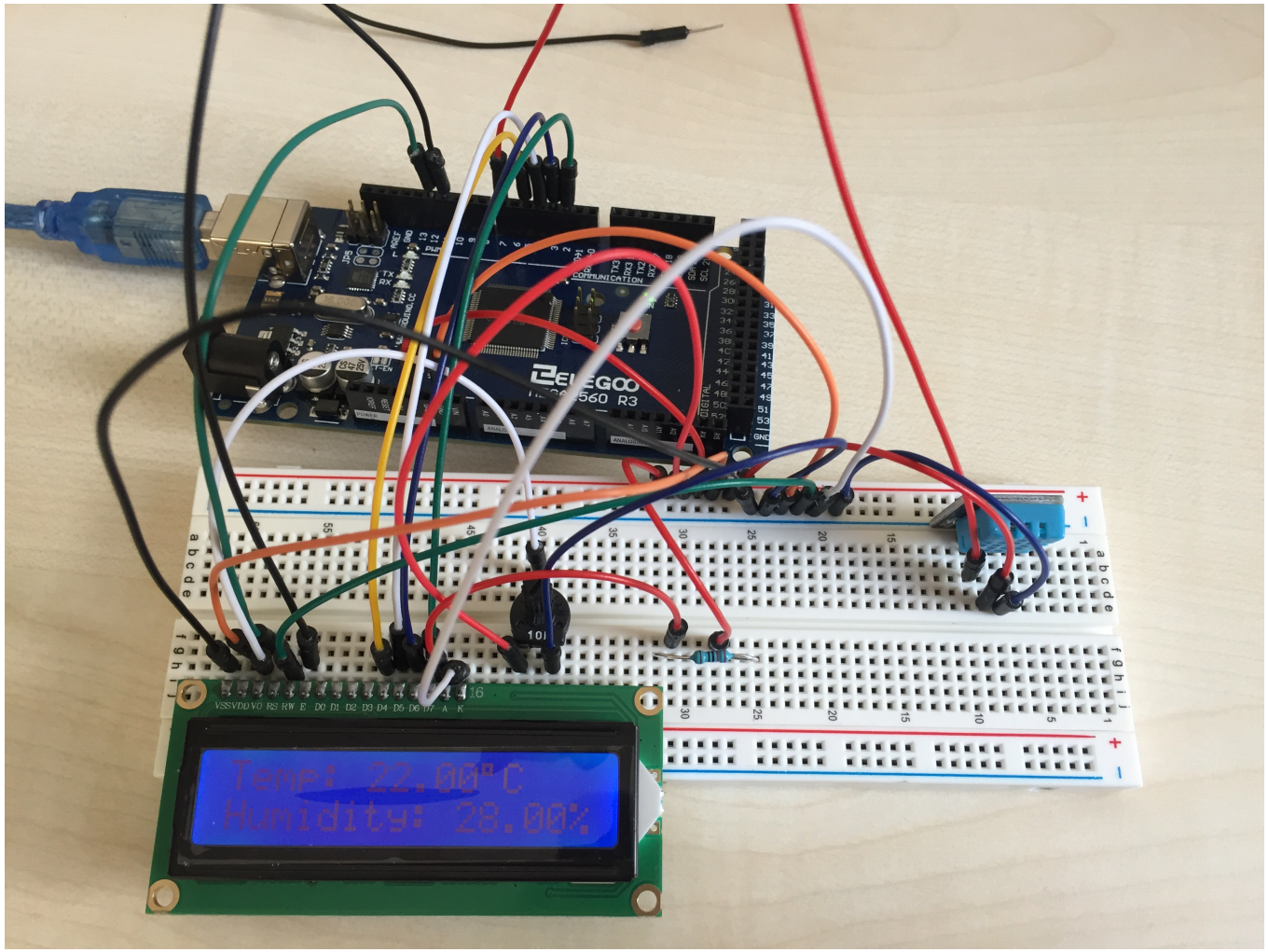


Figure: Fertiges Beispiel

System mit Sensor und LCD

- Übungsaufgabe:

  Tauschen Sie das LCD mit einem *Fan* (Lüfter), der bei einer Temperatur $> 25°C$ oder einer Luftfeuchtigkeit $> 50\%$ aktiviert werden soll.

  Hinweis: Sie benötigen den $L293D$ Chip um den Motor ansteuern zu können.

Einführung: Operational Amplifiers

Junjie Shi Niklas Ueter

LS 12, TU Dortmund

17,April,2019

Operational Amplifier – idealisiertes Modell

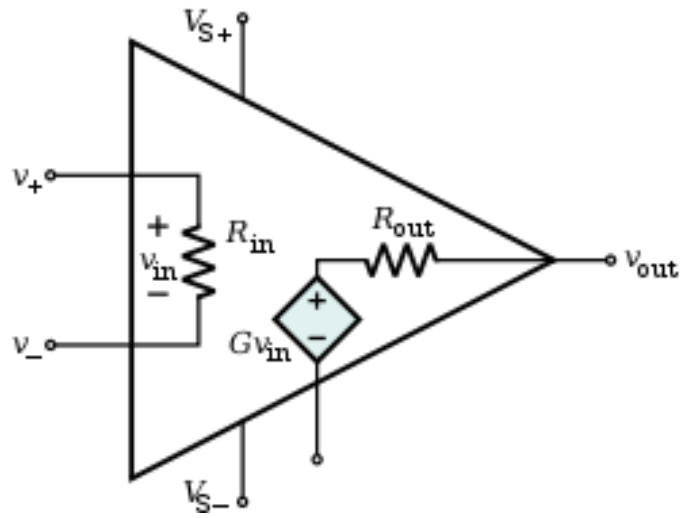


Figure: Modell eines idealen Operationsverstärkers

- Invertierender Eingang V_+, Nicht-invertierender Eingang V_-, Spannungsversorgung V_{S+}, V_{S-}
- Voltage-Controlled-Voltage-Source (VCVS) mit Gain $G \to \infty$
- Innenwiderstand $R_{in} \to \infty$
- Ausgangswiderstand $R_{out} \to 0$

Operational Amplifier – idealisiertes Modell

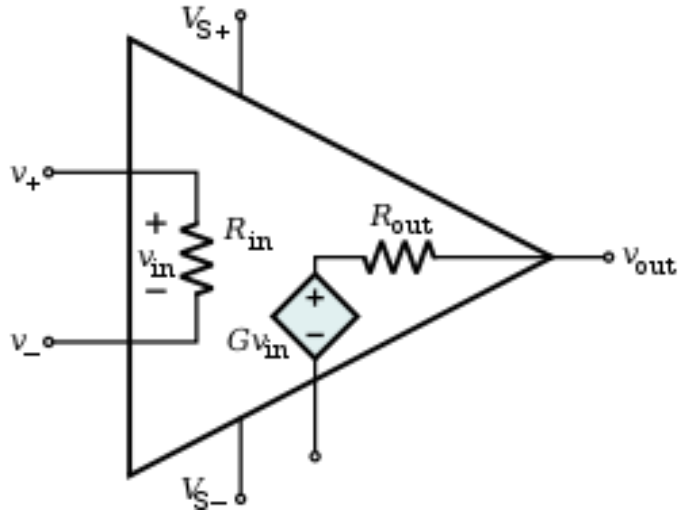
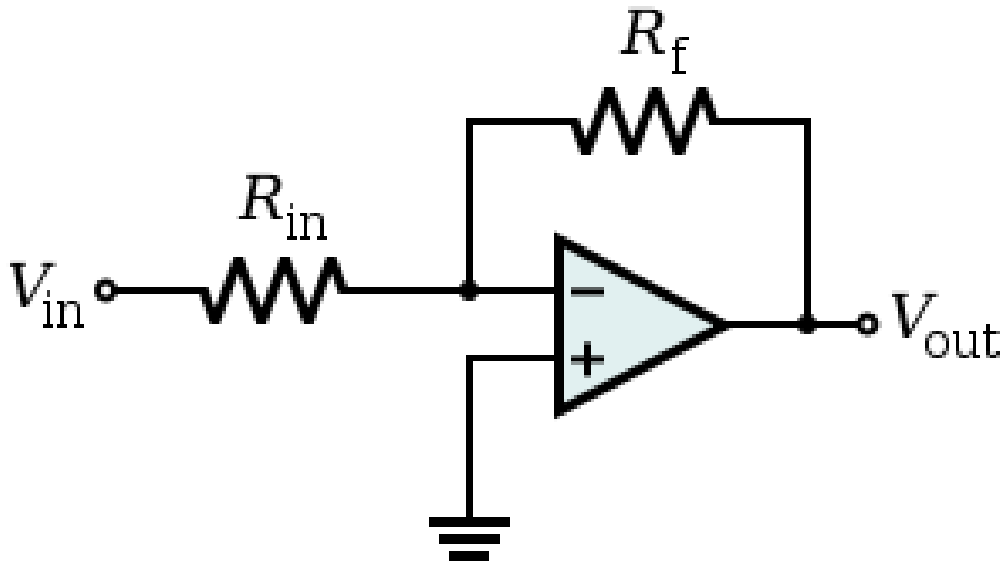


Figure: Modell eines idealen Operationsverstärkers

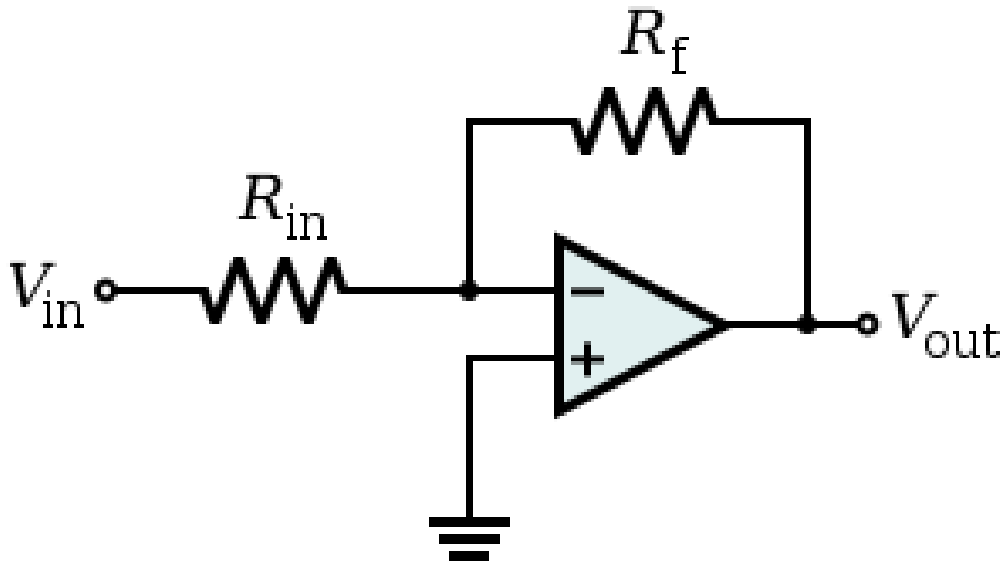
- Zwei abgeleitete 'Rechenregeln':
  - Wegen $R_{in} = \infty \Rightarrow$ Es fließt kein Strom von V_+ zu V_-
  - Im *steady-state* gilt $V_+ = V_-$

Operational Amplifier – Beispielrechnung 1



<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

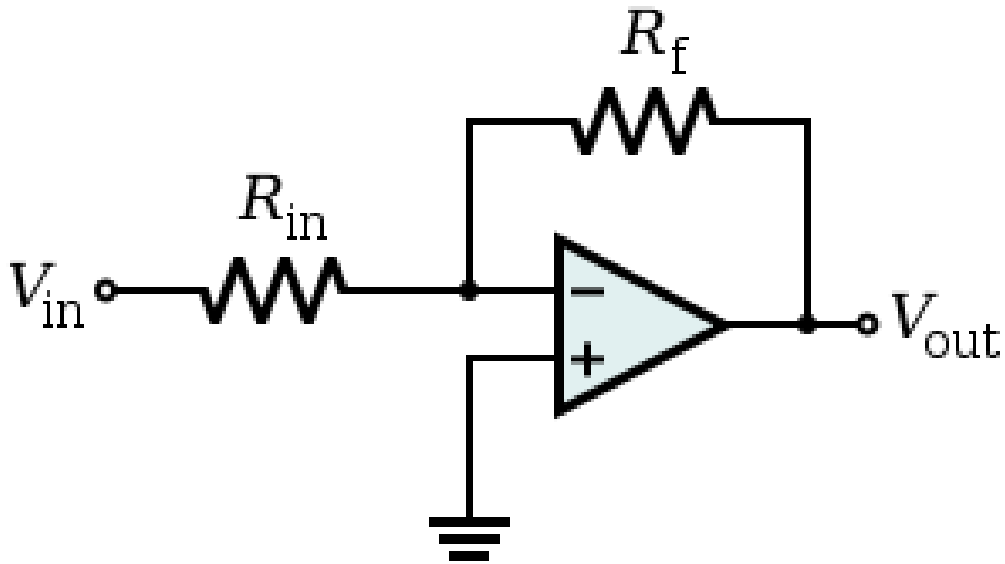
Operational Amplifier – Beispielrechnung 1



<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

(a) $V_- = V_+ = 0$

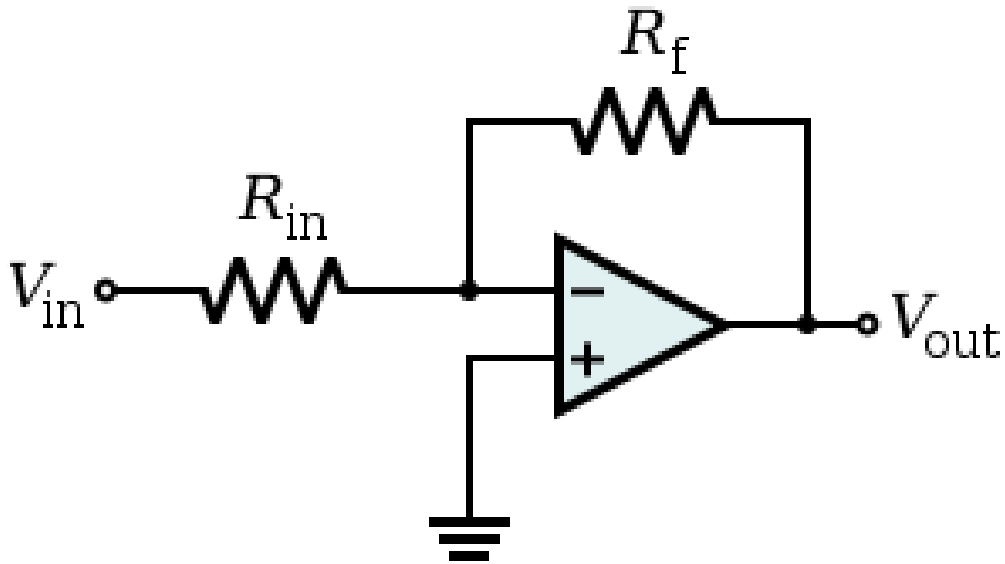
Operational Amplifier – Beispielrechnung 1



<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

(a) $V_- = V_+ = 0$
(b) $I_{R_{in}} = I_{R_f}$

Operational Amplifier – Beispielrechnung 1

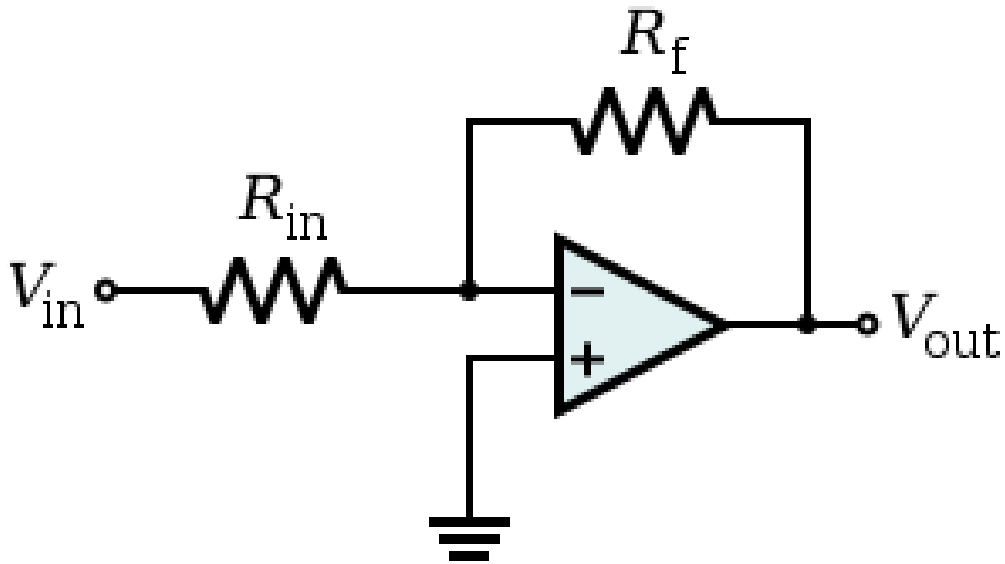


<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

(a) $V_- = V_+ = 0$

(b) $I_{R_{in}} = I_{R_f} \rightarrow (V_{in} - V_-)/R_{in} = (V_- - V_{out})/R_f$

Operational Amplifier – Beispielrechnung 1



<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

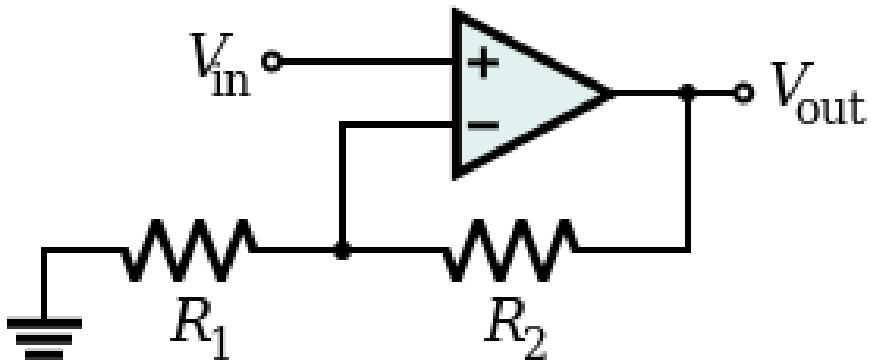
(a) $V_- = V_+ = 0$

(b) $I_{R_{in}} = I_{R_f} \rightarrow (V_{in} - V_-)/R_{in} = (V_- - V_{out})/R_f$

Result: $V_{out} = -(R_f/R_{in}) * V_{in}$

Operational Amplifier – Beispielrechnung 2

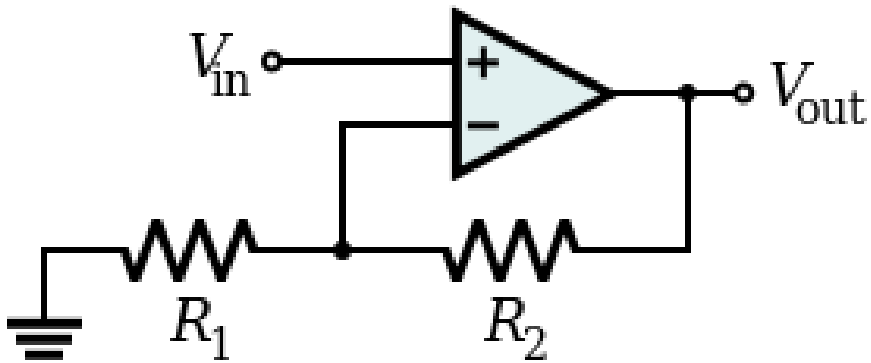
- **Non-inverting amplifier:** Ausgangsspannung 'folgt' der Eingangsspannung



<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

Operational Amplifier – Beispielrechnung 2

- **Non-inverting amplifier:** Ausgangsspannung 'folgt' der Eingangsspannung

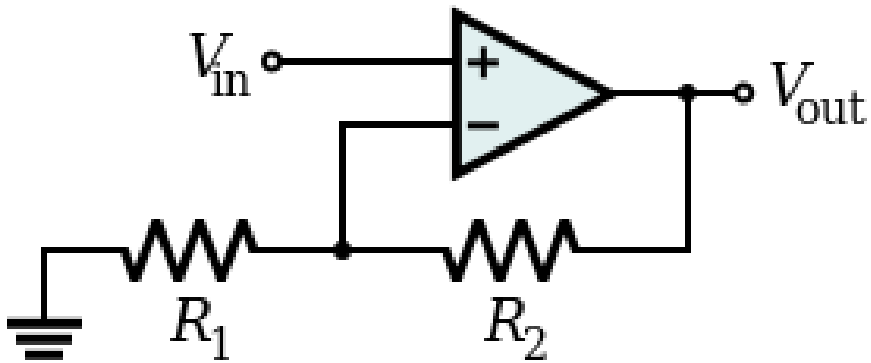


<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

Übung: Bestimmen Sie V_{out}?

Operational Amplifier – Beispielrechnung 2

- **Non-inverting amplifier:** Ausgangsspannung 'folgt' der Eingangsspannung



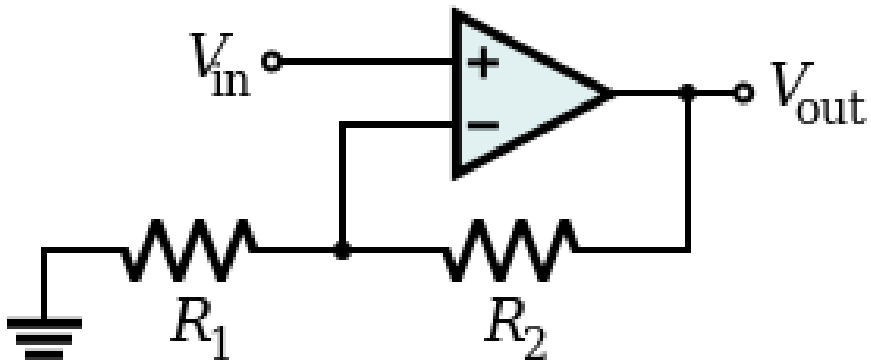
<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

Übung: Bestimmen Sie V_{out}?

Ergebnis: $V_{out} = (1 + R_2/R_1) \cdot V_{in}$

Operational Amplifier – Beispielrechnung 2

- **Non-inverting amplifier:** Ausgangsspannung 'folgt' der Eingangsspannung



<ref: https://en.wikipedia.org/wiki/Operational_amplifier>

Übung: Bestimmen Sie V_{out}?

Ergebnis: $V_{out} = (1 + R_2/R_1) \cdot V_{in}$

Normally, $R_2 \gg R_1$, so $V_{out} \approx (R_2/R_1) \cdot V_{in}$

Weitere Op-Amp Schaltungen

- Differential Amplifiers
- Differenzierer und Integrierer
- Addierer and Subtrahierer
- Analoge Regler (PID)
- Aktive Filterschaltungen
- A/D converter and D/A converter

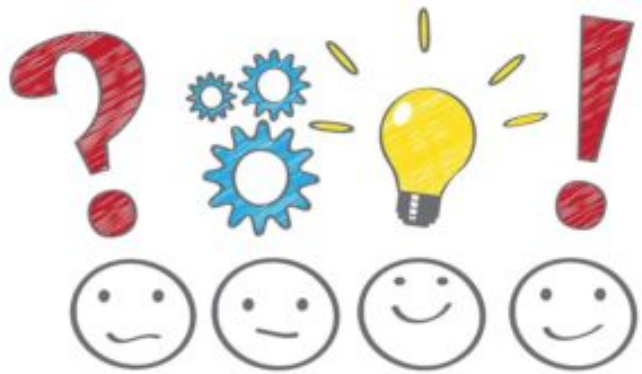
Vorgehen bei realen OpAmps?

1. Berechnung mit dem idealen Modell
2. Welche Abweichungen zum idealen Modell müssen berücksichtigt werden?
3. Beispiel: Gain hat Tiefpass Charakteristik, Slew Rate (maximale erzielbare Steigung des Signals)
4. Nicht-Idealitäten zu gross? Einfluss durch Schaltungstechnische Massnahmen reduzieren

Da die Schaltungen eventuell nicht für alle Projekte interessant sind $\Rightarrow$ individuelle Besprechung bei Bedarf

Fragen

Weitere Fragen?



Tutorial: Abtastung & Quantisierung

Junjie Shi Niklas Ueter

LS 12, TU Dortmund

17,April,2019

Analog-Digital Wandlung – Schritte

1. Analoges Signal wird zeitlich abgetastet – Signalverlauf jedes Abtastintervalls der Dauer T_s wird durch einen Signalwert repräsentiert
2. Quantisierung der wertekontinuierlichen Signalwerte in M verschiedene Amplitudenstufen
   - **Einfach**: Lineare Quantisierung – Abstand zwischen zwei Quantisierungsstufen ist konstant
3. Codierung der Amplitudenstufen in ein Binärmuster

Idealer Abtaster

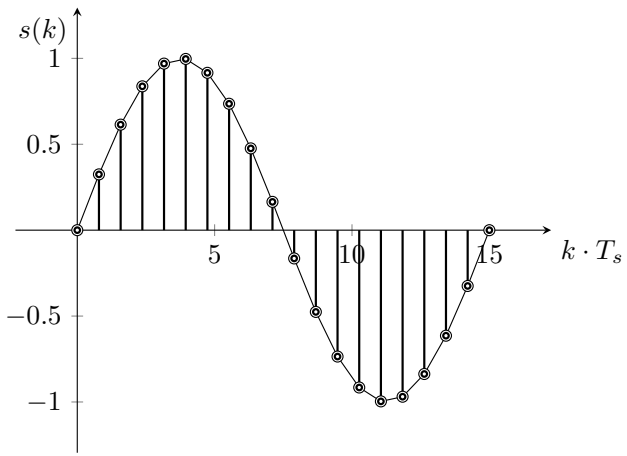
- Abtaster: Aus einem Signal an äquidistanten Stützstellen den Signalwert entnehmen

$\Rightarrow$ Entnimmt zu den äquidistanten Zeitpunkten kT_s den exakten Wert des Eingangssignals $s(kT_s)$

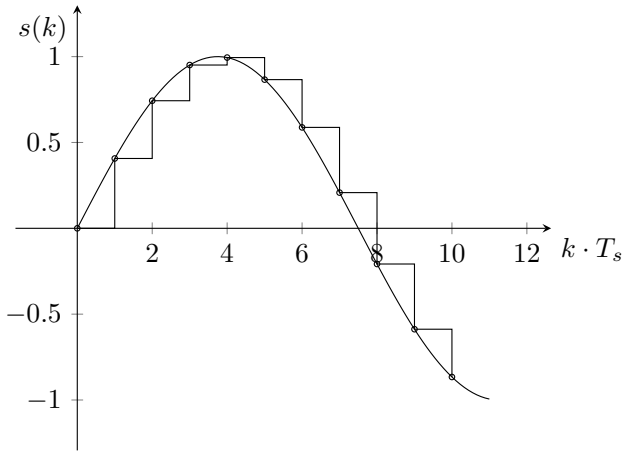
Das abgetastete Signal $s(t)$ wird entsprechend durch die Folge

$$(s(k))_k = \{s(k \cdot T_s) \mid k \in \mathbb{Z}\}$$

repräsentiert



Realer Abtaster



- Natürlich **nicht möglich** den Signalwert an den Stellen kT_s genau zu messen!
- ***Aperture*** $h_A(t)$ bestimmt wie das Signal abgetastet wird – zum Beispiel Rechteckfunktion $\Pi_{T_s}(t)$
- Aperturen $h_A(t)$ verschwindet außerhalb von $[kT_s, (k+1)T_s)$

Realer Abtaster

$$s(kT_s) = \int_{-\infty}^{\infty} s(u) \cdot h_A(u - kT_s)\ du \tag{1}$$

$$s(0) = \frac{1}{T_s} \int_0^{T_s} s(u) \cdot 1\ du,\ s(1) = \frac{1}{T_s} \int_{T_s}^{2T_s} s(u) \cdot 1\ du, \ldots \tag{2}$$

$$s(kT_s) = \int_{-\infty}^{\infty} s(t) \cdot h_A(u-t)\ du\ |_{t=kT_s} = s(t) * h_A(-t)\ |_{t=kT_s} \tag{3}$$

Reale Abtastung verhält sich wie Filterung und anschließende ideale Abtastung

Ideale Abtastung – Dirac-Distribution

- Ideale Apertur ist *unendlich schmal* – z.B. Grenzwert der Rechteckaperture

$$\lim_{T \to \infty} \frac{1}{T} \Pi_T(t), \text{ wobei } \int_{-\infty}^{\infty} \frac{1}{T} \Pi_T(u) \, du = 1$$

$\Rightarrow$

$$\delta(t) = \begin{cases} \textit{undefiniert} & x = 0 \\ 0 & \textit{sonst} \end{cases}, \text{ wobei } \int_{-\infty}^{\infty} \delta(u) \, du = 1$$

Mathematisch *unsauber* aber für die Intuition genug!

Ideale Abtastung – Darstellung im Frequenzbereich

- Dirac-Kamm $\sum_{k=-\infty}^{k=\infty} \delta(t - kT_s)$ ist eine periodische Funktion
  $\Rightarrow$ Fourier-Reihenentwicklung (garnicht so einfach :))

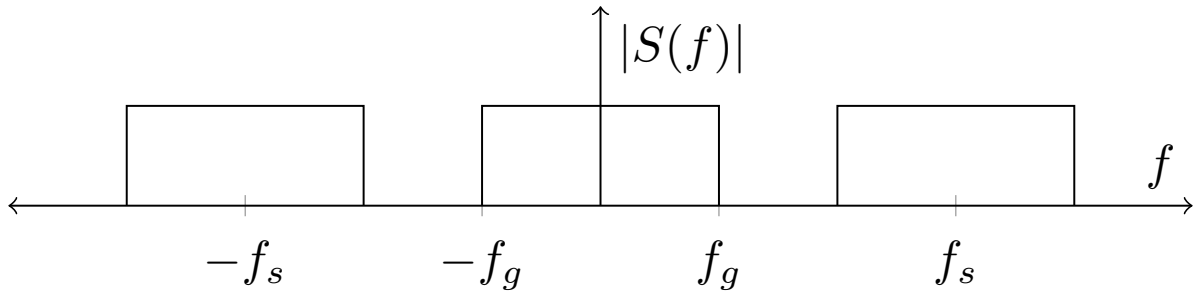
$$\sum_{k=-\infty}^{k=\infty} \delta(f - \frac{k}{T_s})$$

$\Rightarrow \frac{1}{T_s}$ periodische Wiederholung des Spektrums

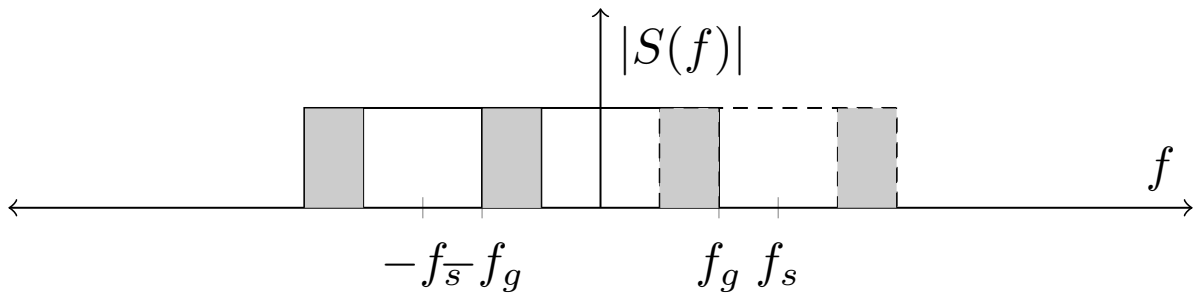
$$\mathcal{F}\left\{ s(t) \cdot \sum_{k=-\infty}^{k=\infty} \delta(t - kT_s) \right\} = S(f) * \sum_{k=-\infty}^{k=\infty} \delta(f - \frac{k}{T_s})$$

$$= \sum_{k=-\infty}^{k=\infty} S(f - \frac{k}{T_s})$$

Abtastung – Darstellung im Frequenzbereich



- Periodische Fortsetzung des Signalspektrums S(f)
- $\Rightarrow fg < f_s - fg$ **sonst** Aliasing



Quantisierung

- Pulse Code Modulation (PCM) – Eingangssignal wird direkt quantisiert und codiert
- Bei der Quantisierung wird nicht jeder beliebige Amplitudenwert, sondern nur noch eine begrenzte Anzahl definierter Ersatzwerte oder auch diskrete Datenwerte d_i zugelassen
- Quantisierungskennlinie gibt Zuordnung an
- Kontinuierliche Spannungswerte, die zwischen zwei Entscheidungsschwellen oder auch Rundungsgrenzen werden einem Ersatzwert zugeordnet $[r_i, r_{i+1}] \mapsto d_i$
- Abstand der Ersatzwert ist Quantisierungsstufe Δ

Quantisierungsrauschen

- Da ganze Wertebereiche eines Signals auf den selben Ersatzwert abgebildet werden $[r_i, r_{i+1}) \mapsto d_i$ kommt es zu Diskretisierungsfehlern (*Quantisierungsrauschen*)
- Können die Signalqualität massiv beeinflussen
- Üblich Signalqualität in $SNR = 10 \cdot log_{10}\left(\frac{P_s}{P_n}\right)$ anzugeben

Minimales Digital-Signalverarbeitendes System

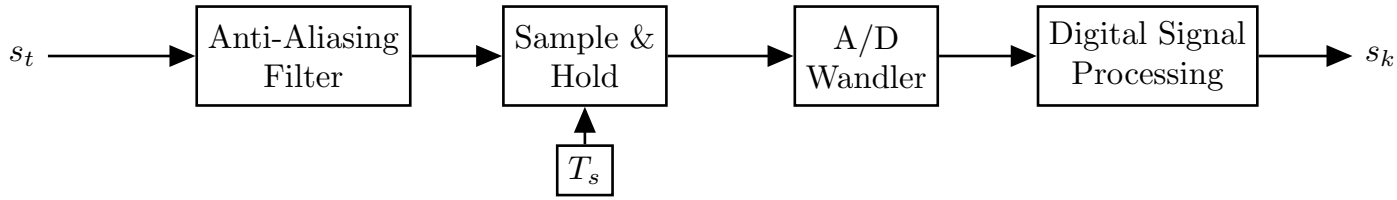
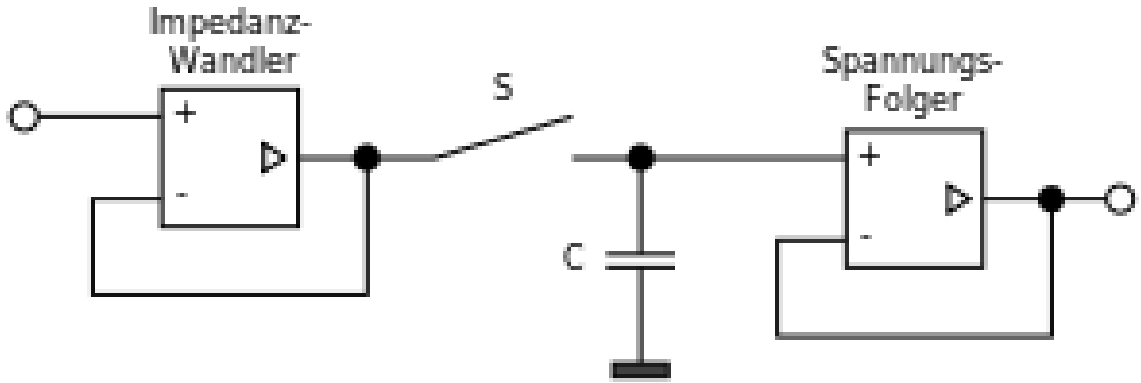


Figure: Blockdiagramm

1. Tiefpass-Filterung (Analog, $f_s/2$)
2. Sample & Hold = Analog Wert (ideal) messen und halten + nicht-ideale Verzerrung
3. Encoding eines konstanten analogen Wertes in eine binäre Repräsentation

Sample & Hold



- Spannung wird $\sim$ konstant gehalten
- Operationsverstärker kompensieren: Elektrische Verluste & Unterschiedliche Widerstände
- In dieser (Hold-) Zeit muss der analoge Wert umgewandelt werden

A/D Wandler

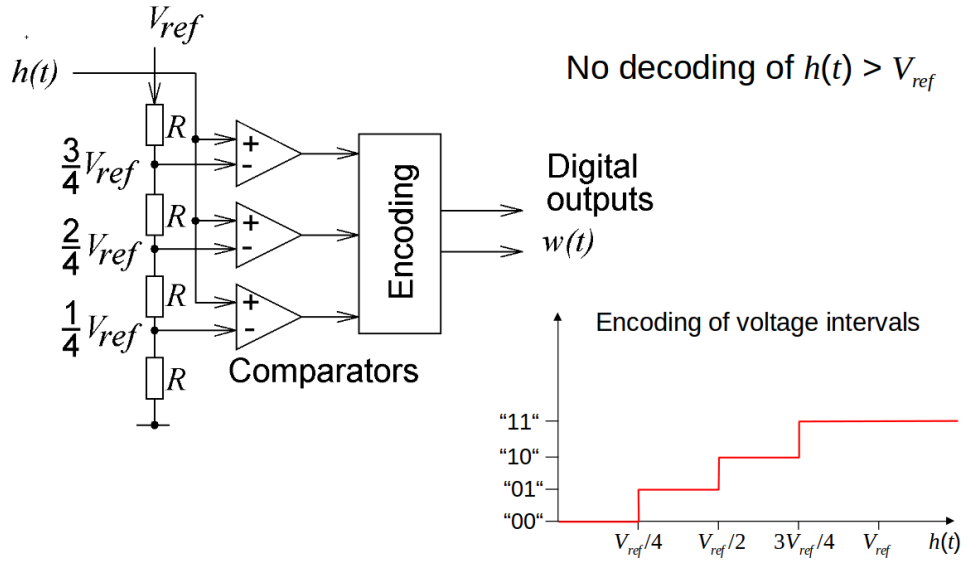


Figure: Flash ADC P.Marwedel and JJ.Chen: Cyber Physical System Fundamental slides 3.1

- **Auflösung:** Die Auflösung des Wandlers bezeichnet die Anzahl an diskreten Spannungswerten, die unterschieden werden können (bezogen auf einen Spannungsbereich).
- $Q = \frac{V_{max}}{2^M}$, wobei M die Anzahl der bits und V_{max} den maximal messbaren Spannungswert bezeichnet.

ADC – Arduino UNO

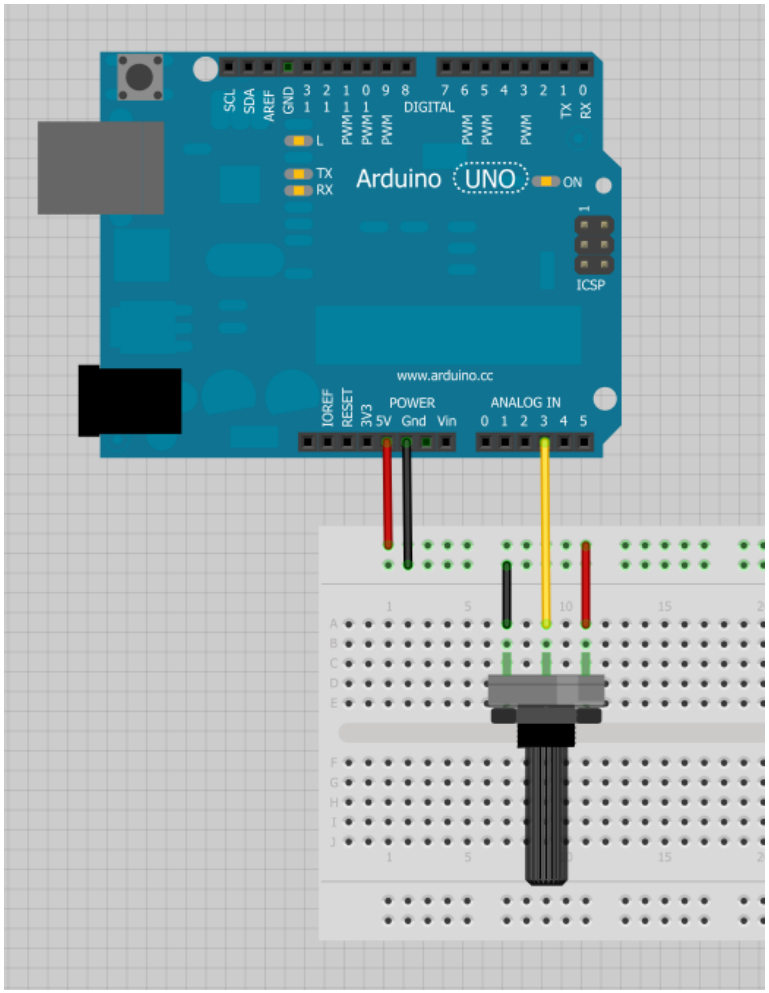


Figure: Ref: learn.sparkfun.com

- 10-Bit ADC
- Lineare Quantisierung
- *Abtasten* der Spannung am Potentiometer
- $5V$ Referenzspannung
- $0x3FF = (1023)_{10} = 5\text{V}$
- $\frac{5V}{1023\ Stufen} \cdot (X)_{10} = Y_{10}[V]$

Beispielcode I

```
void setup() {
   pinMode(A3, INPUT);
}
void loop() {
   int measured = analogRead(A3);
   Serial.print("Measured: ");
   Serial.println(measured);
    delay(2000);
}
```

ref: https://learn.sparkfun.com/tutorials/analog-to-digital-conversion