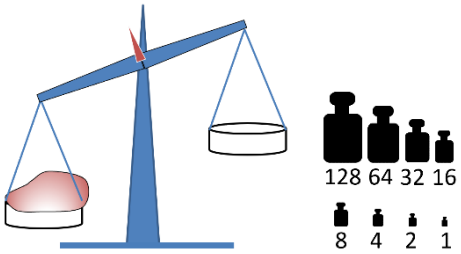
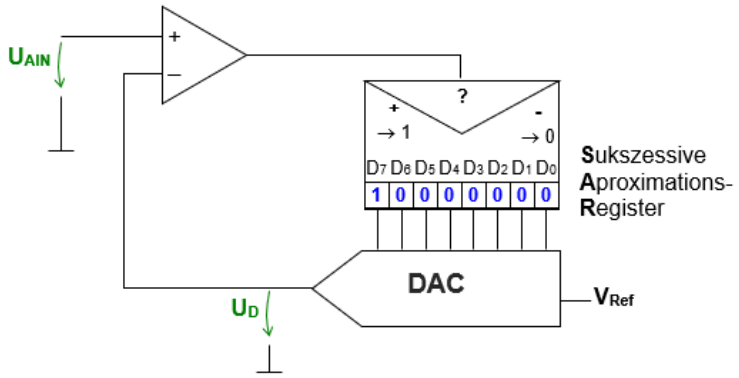
Umwandlung analoger Spannungswerte

Um analoge Spannungen im Mikrocontroller verarbeiten zu können, müssen die Signale in einen digitalen Zahlenwert gewandelt werden. Ein im Mikrocontroller häufig angewendetes Verfahren ist das Wägeverfahren, oder Verfahren der schrittweisen Annäherung (Sukszessive Approximation). Dabei erfolgt sozusagen ein digitaler Wägevorgang, den man mit dem systematischen Wiegen einer Ware auf einer Balkenwaage vergleichen kann:



Ist das Gewicht der Ware unbekannt, legt man zunächst das größte Gewicht (128 =2^7) in die Waagschale. Schlägt der Zeiger nach rechts aus, muss das Gewicht wieder runtergenommen werden. Ansonsten verbleibt es in der Schale. Danach wird das nächst kleinere Gewicht (64 = 2^6) auf-, bzw. dazugelegt.
Dieser Vorgang wiederholt sich, bis (hier) nach 8 Wägeschritten alle Gewichte durchprobiert sind.

Beim AD-Wandler ist die „Ware“ die Analogspannung (U_{AIN}). Die Rolle der Waagschale übernimmt ein Digital-Analog-Wandler (DAU oder DAC) und die „Gewichte“ sind die mit der jeweiligen Wertigkeit versehenen Bit D0 … D7 des SAR. Der ADC erzeugt daraus eine analoge Vergleichsspannung (U_D). Als „Zeiger“, der nach links oder rechts ausschlägt, fungiert der Komparator. Abhängig vom Vergleichsergebnis legt die Steuerung die „Gewichte“, beginnend mit D7, auf. Auch hier ist die Messung nach 8 Schritten beendet.



Die Größe der Gewichte richtet sich nach der **Spannungsauflösung** des ADU. Bei einem Messbereich von 5V beträgt das kleinste „Gewicht“, bzw. der kleinstmögliche Spannungsschritt U_{LSB}:

$$U_{LSB} = \frac{U_{Ref}}{2^n}$$

U_{Ref}: Referenzspannung, oder Messbereichsendwert
n: Anzahl der Bit (man sagt: Der ADU hat n Bit „Auflösung“)

Die Berechnung der Analogspannung erfolgt im Mikrocontroller dann mit:

$$U_{AIN} = \# \cdot U_{LSB}$$

#: Digitaler Zahlenwert mit n Bit Auflösung

Aufgabe

Bei einem 12-Bit ADU soll der anliegende Spannungswert U_{AIN} = 2,953V betragen. Die Referenzspannung beträgt U_{Ref} = 3,3V.

a) Ermitteln Sie U_{LSB}.

b) Beschreiben Sie den Ablauf des Wägevorgangs.

c) Welchen digitalen Zahlenwert erhalten Sie? Geben Sie den Wert als Binär-, Hexadezimal- und Dezimalzahl an.

d) Der ADU ermittelt einen Zahlenwert von # = 3294. Welcher Spannungswert wurde gemessen?

Programmierung des ADU

Zum Lesen eines Analogwerts gibt es einen fertigen Arduino-Befehl:

```
uint16_t wert = analogRead(GPIO);
```

Da der ESP32 eine Auflösung von n = 12 Bit und eine Referenzspannung von 3,3V besitzt, ergibt sich die Spannungsauflösung zu:

$$U_{LSB} = \frac{U_{Ref}}{2^n} = \frac{3{,}3V}{2^{12}} = \frac{3{,}3V}{4096} = 0{,}008057\ V$$

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  uint16_t wert = analogRead(34);
  Serial.print(wert);
  Serial.print(" -> ");

  float spannung = 0.0008057 * (float)wert;
  Serial.print(spannung); Serial.println("V");

  delay(50);
}
```

Damit kann der Analogwert gelesen und über den seriellen Monitor als Spannungswert ausgegeben werden.

Die Ausgabe erfolgt als Gleitkommazahl mit einfacher Genauigkeit.
Vor der Multiplikation mit U_{LSB} wird der eingelesene Wert in den Datentyp **float** umgewandelt („gecastet“).

Arbeitsaufträge

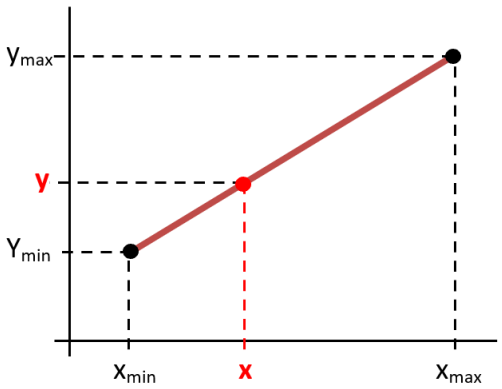
1. Programmieren und testen Sie den abgebildeten Programmcode. Analysieren Sie das Programm.

2. Das ESP32-Carrier-Board hat über Spannungsteilung einen 5V-Messbereich. Beim Anlegen von 5V (z.B. Poti Rechtsanschlag) wird im seriellen Monitor daher nur 3,3V angezeigt.
   Entwerfen Sie eine allgemeine Funktion zur Messbereichsanpassung (Mapping), mit folgender Funktionsdeklaration:

   ```
   float adcMap(float x, float x_min, float x_max, float y_min, float y_max);
   ```

   Die Umrechnungsformel lässt sich mit Hilfe der nebenstehenden Kennlinie herleiten:

   $$\frac{y - y_{min}}{y_{max} - y_{min}} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

   Stellen Sie die Formel um auf y und erstellen Sie die Funktion. Der Funktionsaufruf sollte dann so aussehen:

   ```
   spannung = adcMap(spannung,0.0,3.3,0.0,5);
   ```

3. Erstellen Sie ein Programm, das mit dem Poti die Blinkfreuquenz der LED (io32) ändert.

Dokumentieren Sie Ihre Ergebnisse.