

Thema: PWM-Signal (Pulsweitenmodulation)

Keywords

- ✓ PYTHON ANACONDA ¹
- ✓ Jupyter Notebook / matplotlib / Erstellung von Diagramme
- ✓ ChatGPT ²
- ✓ PWM-Signal

¹<https://docs.anaconda.com/free/anaconda/>

²<https://chat.openai.com/>

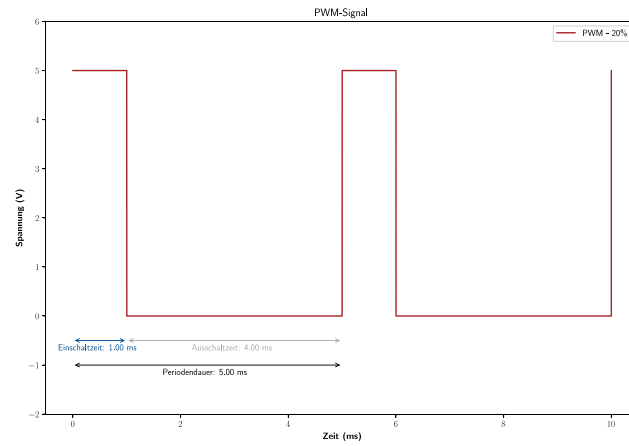


Abb. 1: PWM 20%

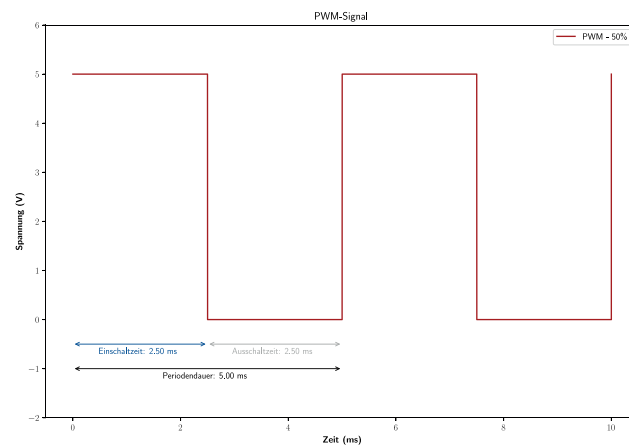


Abb. 2: PWM 50%

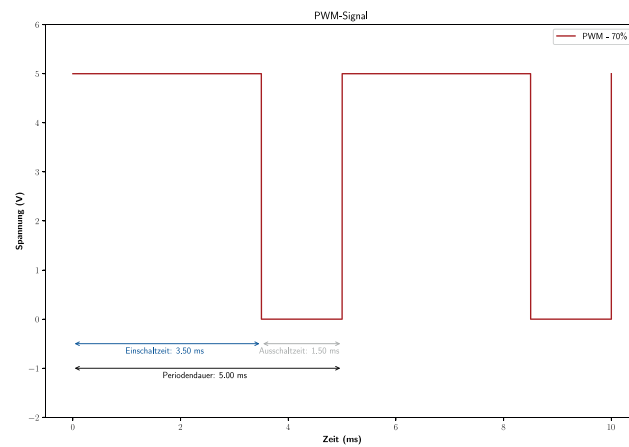


Abb. 3: PWM 70%

```

# Quellcode in Python, PWM-Signal
import math
import matplotlib
matplotlib.rcParams['text.usetex'] = True # Latex code
import numpy as np
import matplotlib.pyplot as plt
# PWM-Parameter
Vmax = 5 # Maximale Spannung
frequenz = 200 # Frequenz in Hz
TV = 0.7 # Tastverhältnis (0.5 entspricht 50%)
sample_rate = 10000000 # Abtastrate
T = 1 / frequenz # Periodendauer
t = np.linspace(0, 2*T, int(2 * T * sample_rate)) # Zeitvektor für 2 Perioden
# PWM-Signal erzeugen
pwm_signal = Vmax * (t % T < T * TV)
# Einschalt- und Ausschaltzeit berechnen
einschaltzeit = TV * T
ausschaltzeit = T - einschaltzeit
# Plot erstellen
# Farbe: Orange #F28C64 grau2 #B2B2B2 rot5 #A71916 blau5 #0D468E
plt.figure(figsize=(250/25.4, 176/25.4)) # Größe in inches (B5 format: 250mm x
    ↳ 176mm)
# Multipliziere t mit 1000, um Zeit in ms zu konvertieren
plt.plot(t*1000, pwm_signal, label=f'PWM - {TV*100:.0f}%\%', color="#A71916")# f<<
    ↳ erlaubt variablen, rot5 #A71916
plt.title(r"PWM-Signal", fontsize=12)# Latex r<<
plt.xlabel(r"\textbf{Zeit (ms)}") # Latex
plt.ylabel(r"\textbf{Spannung (V)}")# Latex
#plt.grid(True)
plt.legend()
# Pfeile und Text hinzufügen
plt.annotate("", xy=(0, -0.5), xytext=(einschaltzeit*1000, -0.5),
    arrowprops=dict(arrowstyle="<->", color='#0D468E'))
plt.text(einschaltzeit*1000/2, -0.7, f"Einschaltzeit: {einschaltzeit*1000:.2f} ms
    ↳ ",
    horizontalalignment='center', color='#0D468E')# blau5 #0D468E
plt.annotate("", xy=(einschaltzeit*1000, -0.5), xytext=(T*1000, -0.5),
    arrowprops=dict(arrowstyle="<->", color='#B2B2B2'))
plt.text(einschaltzeit*1000 + ausschaltzeit*1000/2, -0.7, f"Ausschaltzeit: {
    ↳ ausschaltzeit*1000:.2f} ms",
    horizontalalignment='center', color='#B2B2B2')# grau2 #B2B2B2
plt.annotate("", xy=(0, -1), xytext=(T*1000, -1),
    arrowprops=dict(arrowstyle="<->", color='black'))
plt.text(T*1000/2, -1.2, f"Periodendauer: {T*1000:.2f} ms",
    horizontalalignment='center', color='black')# black
# Platz für Anmerkungen lassen
plt.ylim(-2, Vmax+1)
plt.tight_layout()
plt.savefig("Diag_PWM_70.svg")# SVG-Vektorgrafik
plt.show()

```