

Programmierung von ATMEL AVR Mikroprozessoren am Beispiel des ATtiny13

Eine Einführung in Aufbau, Funktionsweise,
Programmierung und Nutzen von Mikroprozessoren

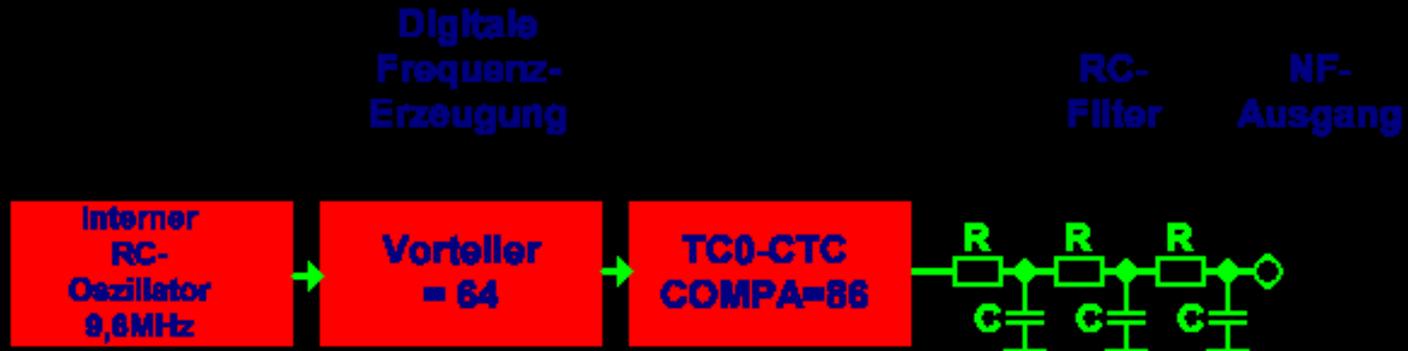


Teil 8: Ein Sinusgenerator mit AVR-Timer

Sinus mit Timer und RC-Filter

Aufgabe: Einen Sinus erzeugen, hier: 1750 Hz

Einfache Lösung: Timer mit CTC, toggeln des OCR0A- oder OR0B-Ausgangs und RC-Filter zum Sieben der Oberwellen nachschalten



$$9,6\text{MHz} / 64 / 86 = 1744,19 \text{ Hz}$$

Sinusgenerator mit Timer/Counter

Nachteile des Verfahrens

- **Schaltfrequenz des Ausgangs =
2 * Ausgangsfrequenz**
- **Erste Oberwelle des Rechteck-Signals =
3 * Grundwelle**
- **Schlechte Dämpfung von RC-Filtern bei
3 * Grundfrequenz**
- **Für ausreichende Dämpfung ist größeres RC-Glied
notwendig**
- **Dadurch schon hohe Dämpfung bei der Nutzfrequenz**

Besser: Sinusgenerator mit PWM

- **Prinzip der Pulsweitenmodulation:**
Die Zeitdauer im High-Zustand und die Zeitdauer im Low-Zustand bei einem Rechteckgenerator sind variabel.
Sind beide gleich, ergibt sich am Ausgang im Mittel die halbe Betriebsspannung.



Prinzip der Pulsweitenmodulation

- Ist der High-Zustand kürzer als der Low-Zustand, ergibt sich eine niedrigere mittlere Spannung:



- Ist der High-Zustand länger als der Low-Zustand, ergibt sich eine höhere mittlere Spannung:



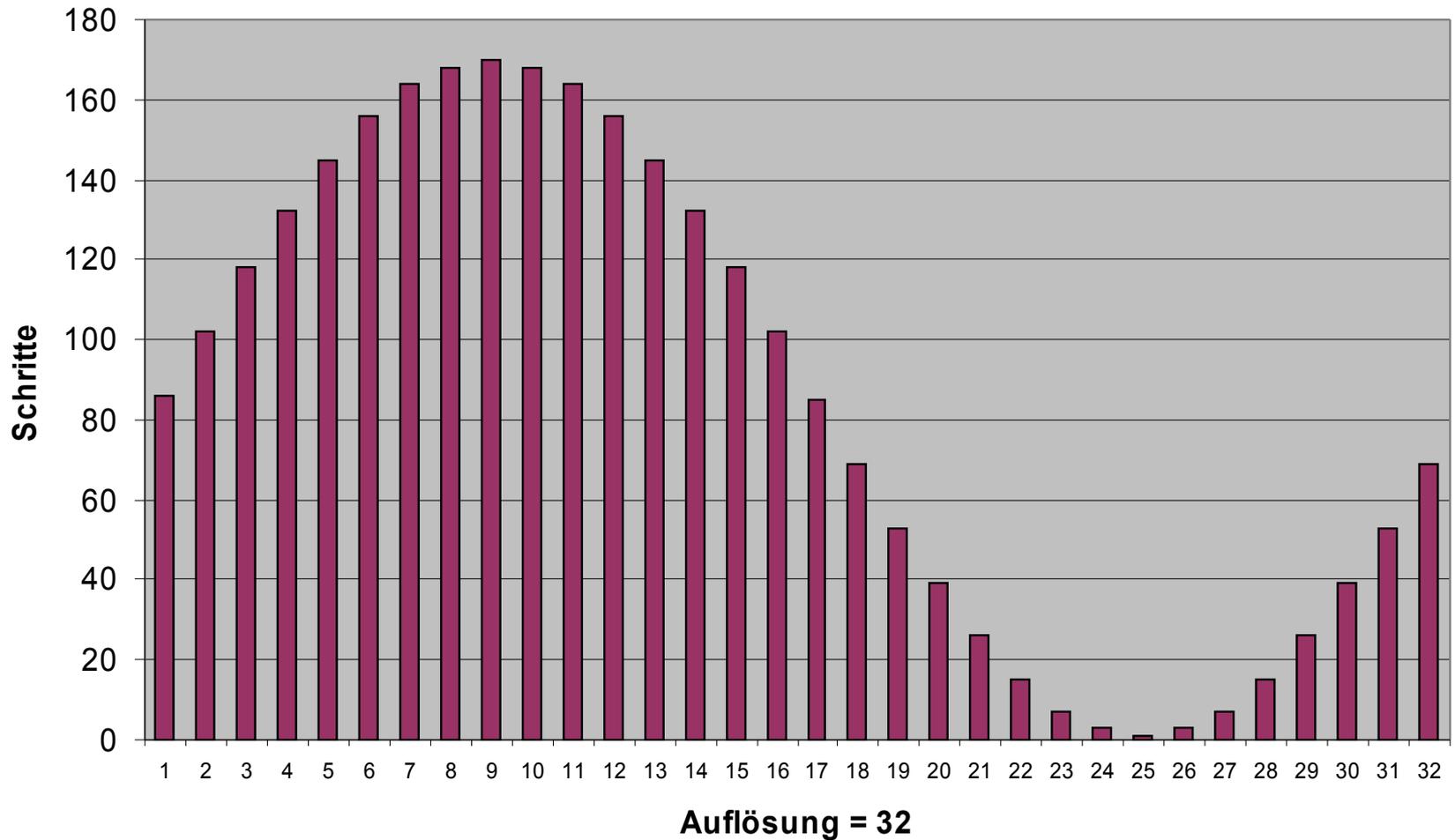
- Mit Pulsweitenmodulation lassen sich analoge Spannungen zwischen Null und Betriebsspannung erzeugen.
- Die Auflösungsgenauigkeit der „Analogspannung“ ist nur von der Schrittzahl abhängig.

Timer als PWM-Generator

- Die Anzahl Schritte der PWM ergibt sich aus dem Taktgenerator (hier: 9,6MHz), der gewünschten Auflösung (hier: 32) und der gewünschten Frequenz (hier: 1750).
$$n = 9.600.000 / 32 / 1750 = 171,43$$
- Anzahl Schritte der PWM wird mit dem CTC-Wert festgelegt (CTC = Clear Timer on Compare). Hier erfolgt das bei 171. Dieser Wert kommt nach Compare Match A.
- Die PWM-Werte müssen zwischen 0 und 171 variieren. Diese Werte kommen nach Compare Match B.
- Compare Match A setzt den Zähler auf Null und den Ausgang auf High. Bei Erreichen von Compare Match B wird der Ausgang Low. Je größer Compare Match B, desto höher die Spannung.

PWM-Werte für einen Sinus

Sinus-PWM mit Schrittweite 171



Auslegung des RC-Filters mit SinePwm

Simulation Sinus mit PWM

Simulation eines Sinusgenerators mit einem PWM

Eingabewerte Frequenzen:

Taktoszillator (MHz): Verteiler: Auflösung: Sollfrequenz (Hz):

Berechnete Werte:

CTC-Wert Soll: CTC-Wert Ist: Frequenz Ist: Abweichg. %:

Empfohlene RC-Werte:

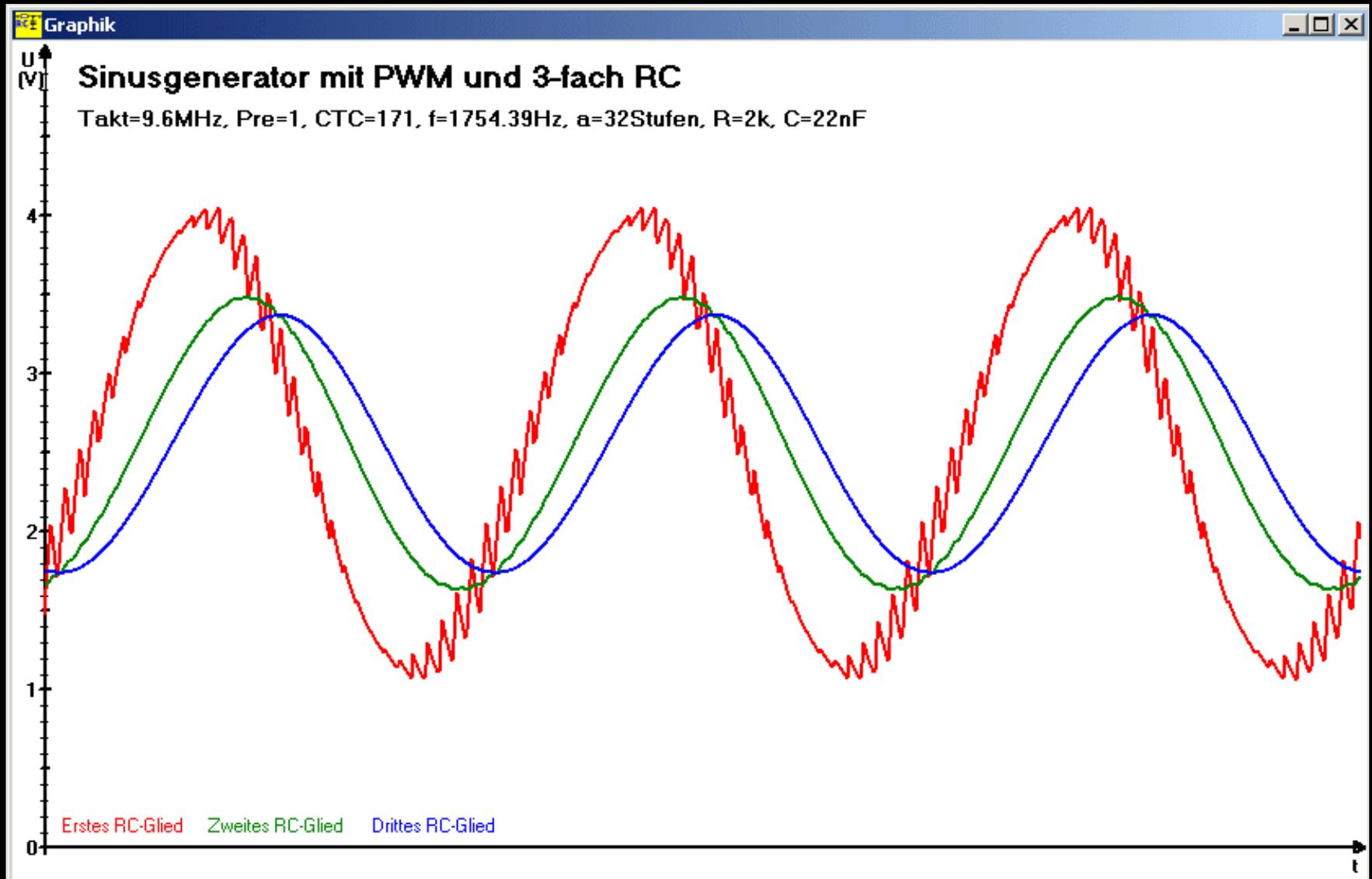
$R \cdot C$ (kOhm*nF) = Widerstand (kOhm): Kondensator (nF):

Gewählte Werte für Simulation 3-fach RC:

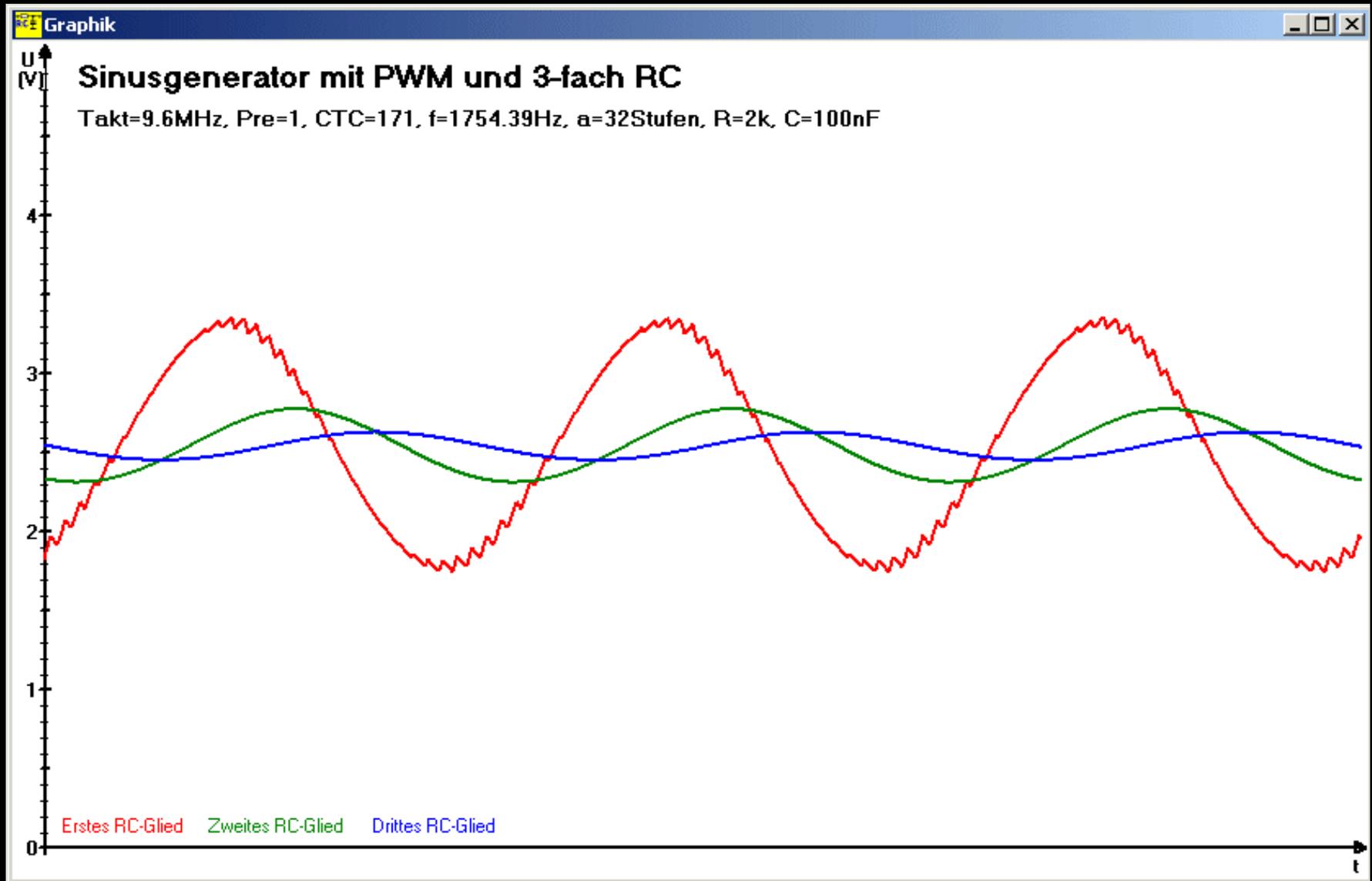
RC-Filter:	Widerstand (kOhm): <input type="text" value="2.2"/>	Kondensator (nF): <input type="text" value="22"/>	<input type="button" value="Bild"/>
PWM-Spannungsbereich:	Minimum (V): <input type="text" value="0.0"/>	Maximum (V): <input type="text" value="5.0"/>	<input type="button" value="GO!"/>
Simulationsauflösung:	Anzahl Schritte: <input type="text" value="1000"/>	Anzahl Vollwellen: <input type="text" value="3"/>	<input type="button" value="Ende"/>

Emitterfolger

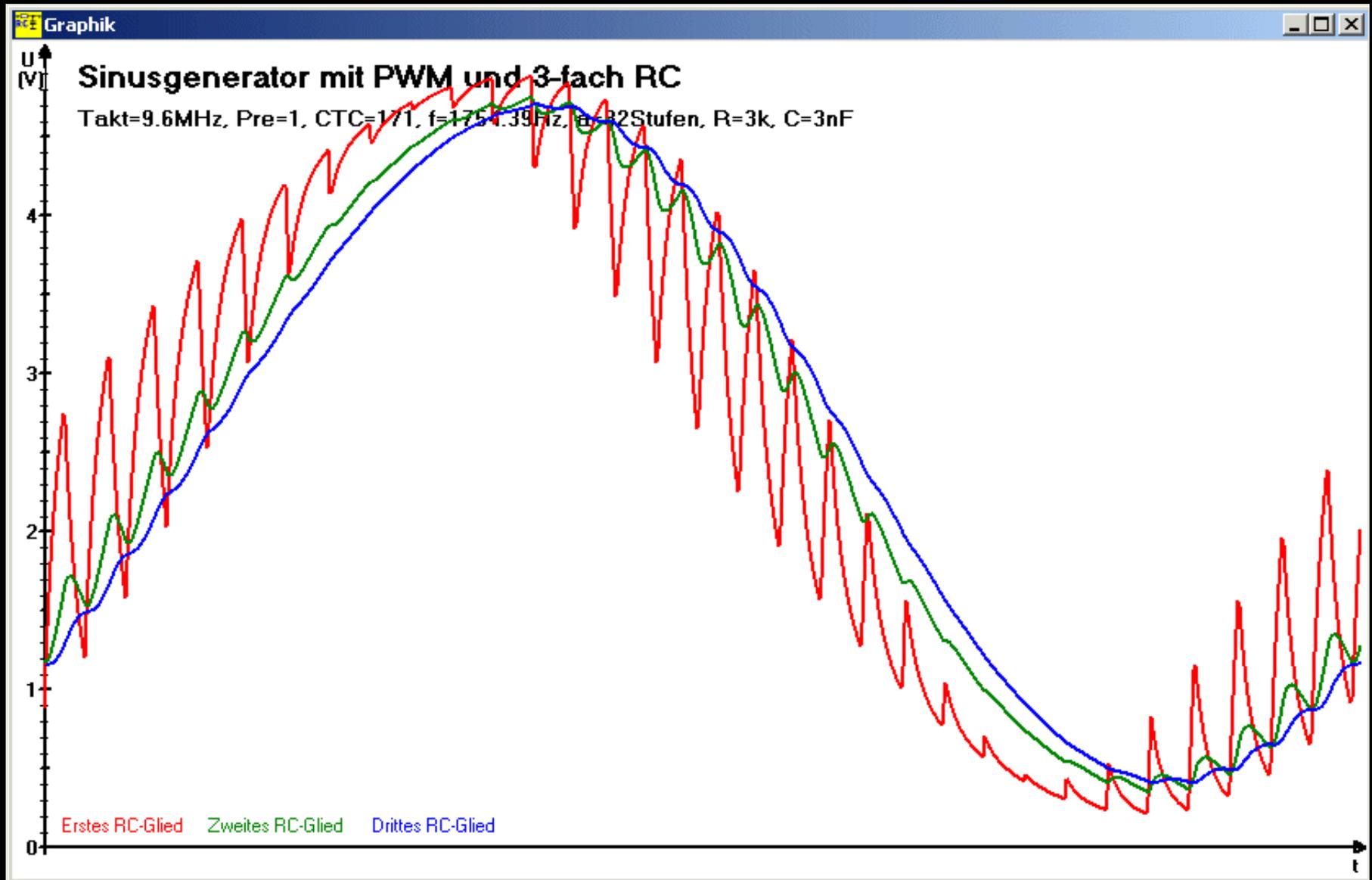
Graphische Ausgabe von SinePwm



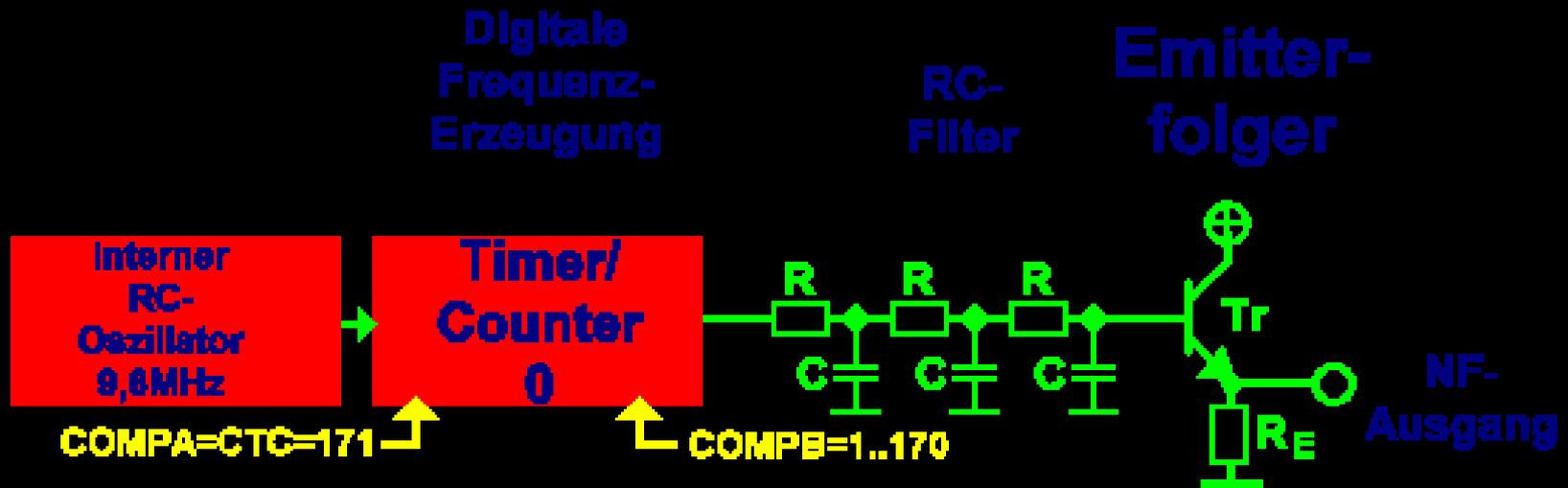
Zu großes RC



Zu kleines RC



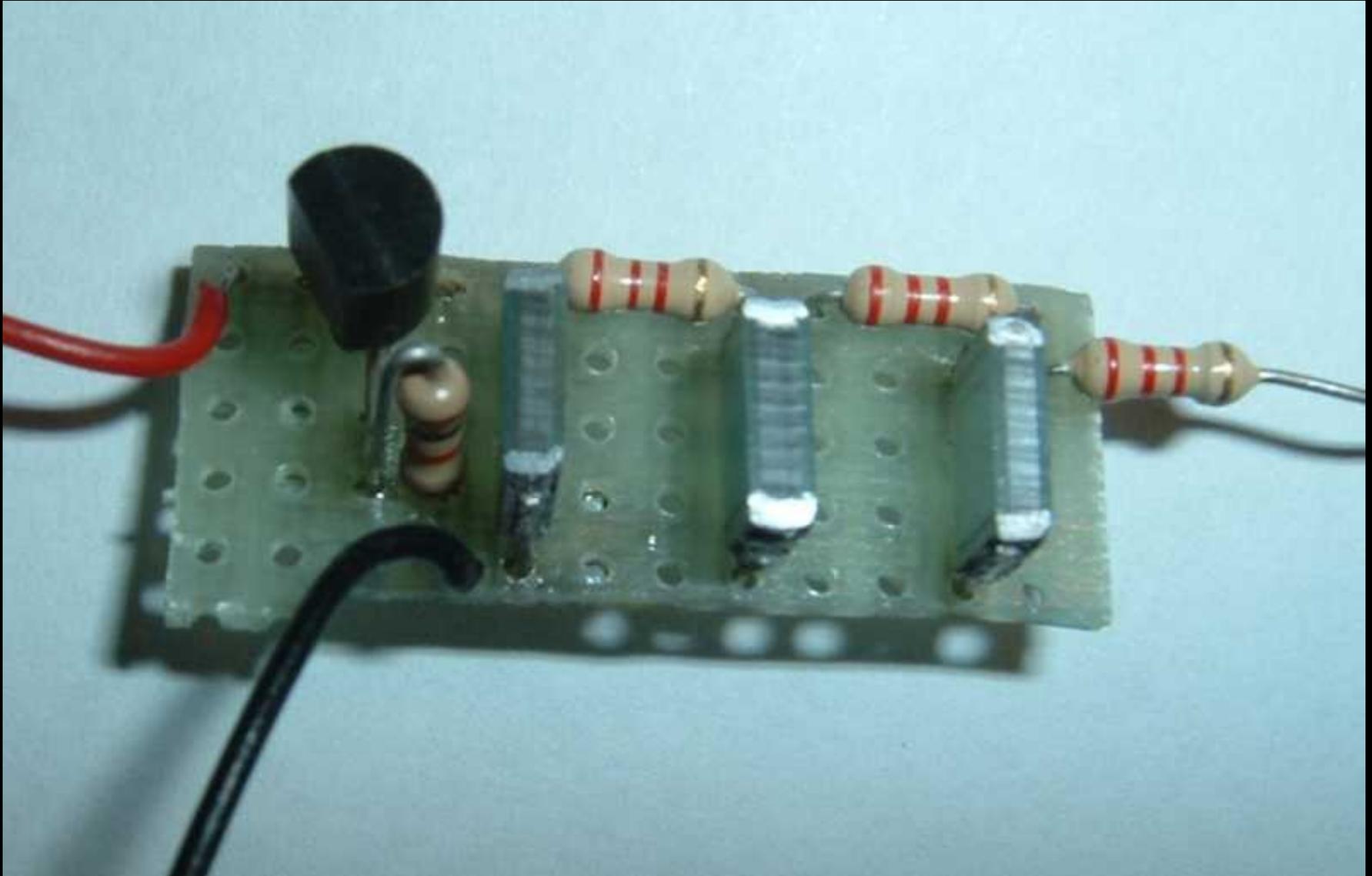
Emitterfolger als Treiber



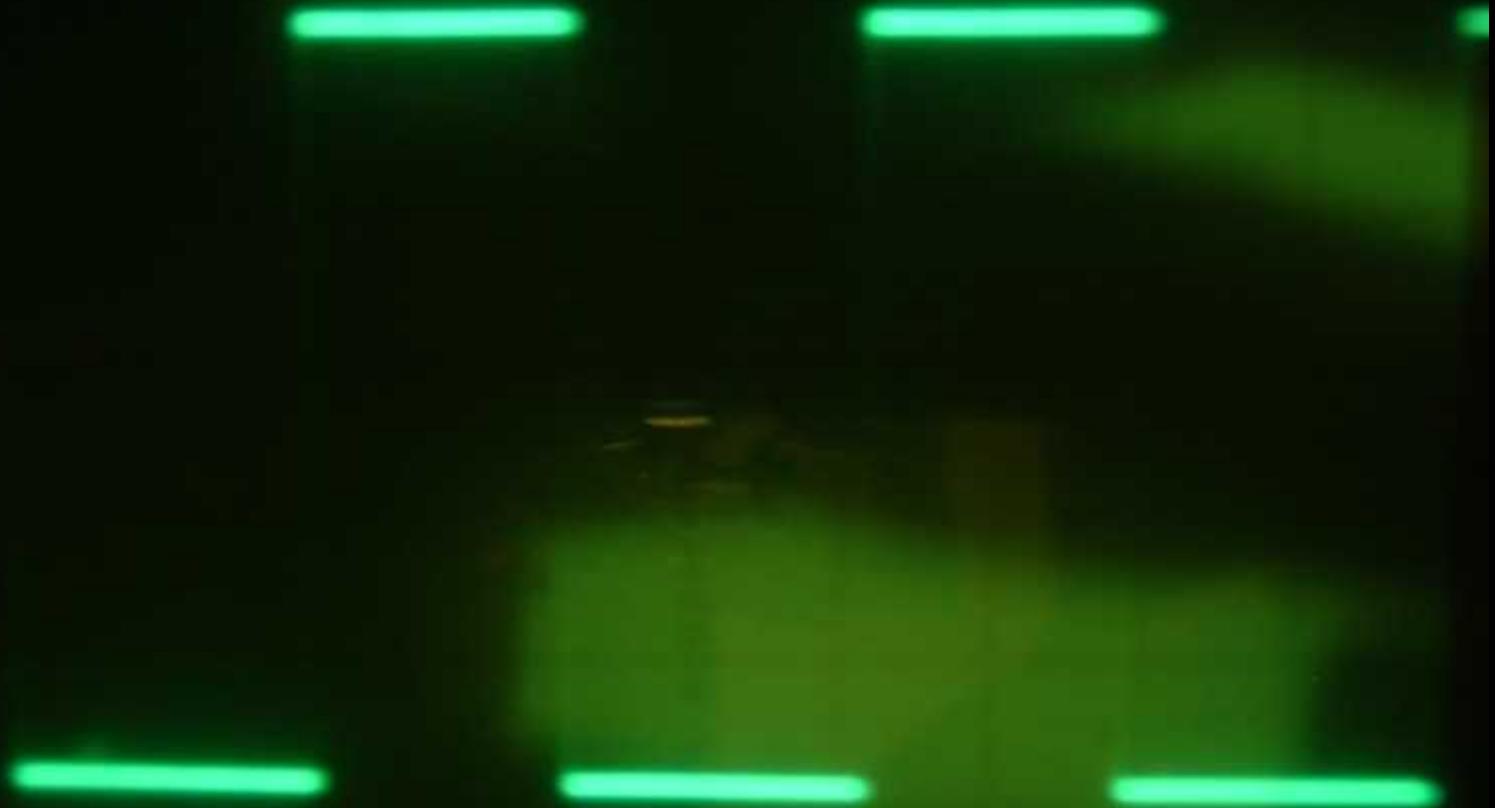
$$9,6\text{MHz} / 171 / 32 = 1754,39\text{Hz}$$

Sinusgenerator mit PWM

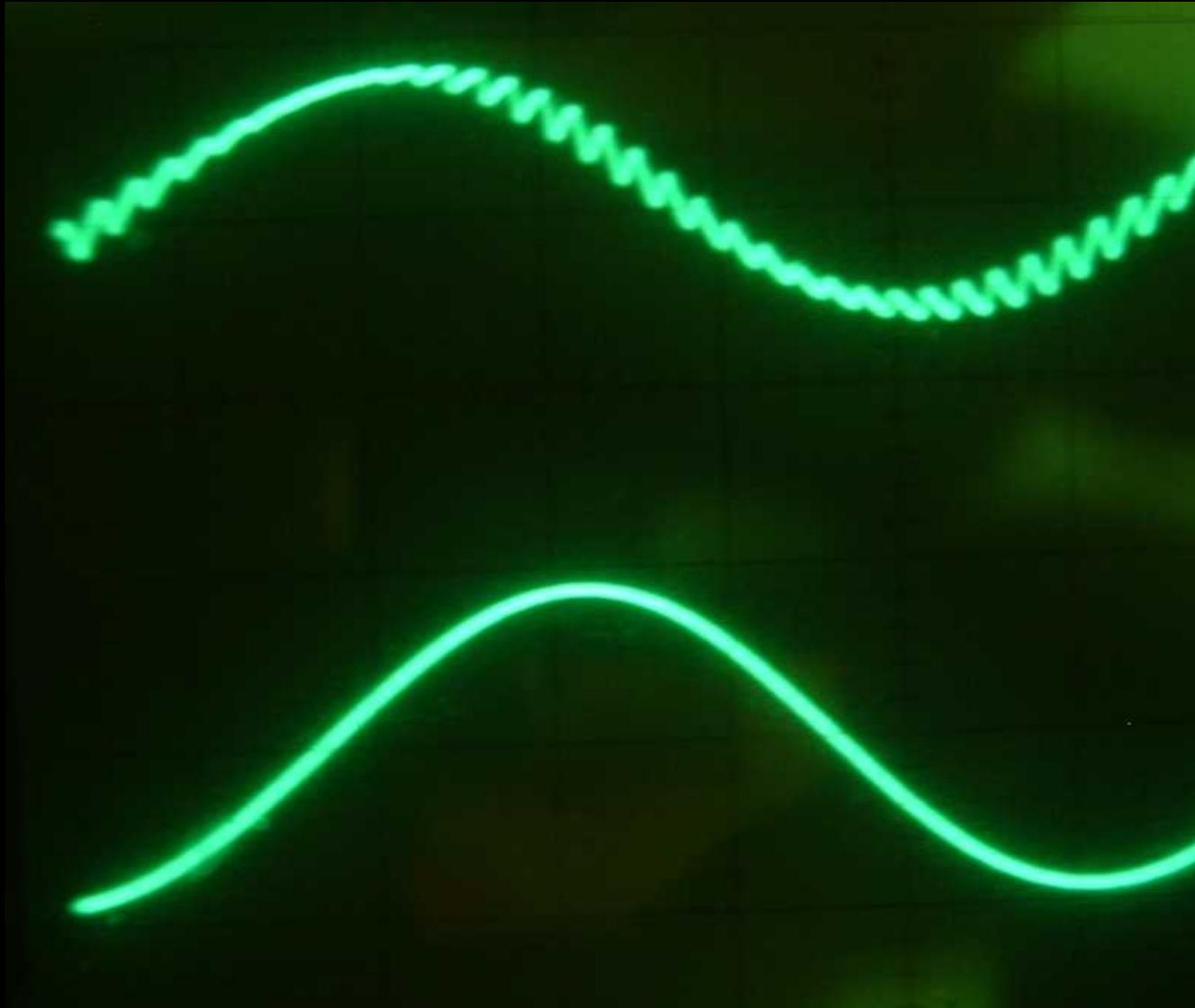
Die Schaltung



Das Ergebnis – PWM-Ausgangssignal



Das Ergebnis – Analog-Ausgangssignal



Oben:

**Erstes RC-
Glied**

**Zweistrah-
oszillo-
gramm**

Unten:

**Emitterfolger-
Ausgang**