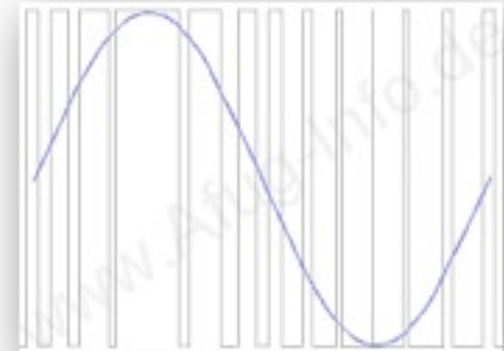
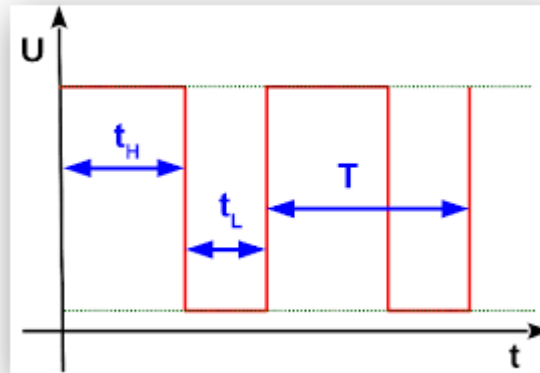
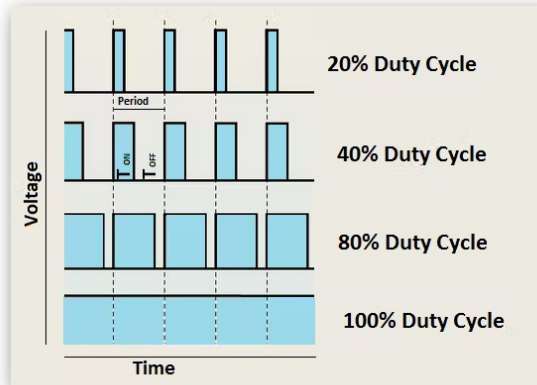


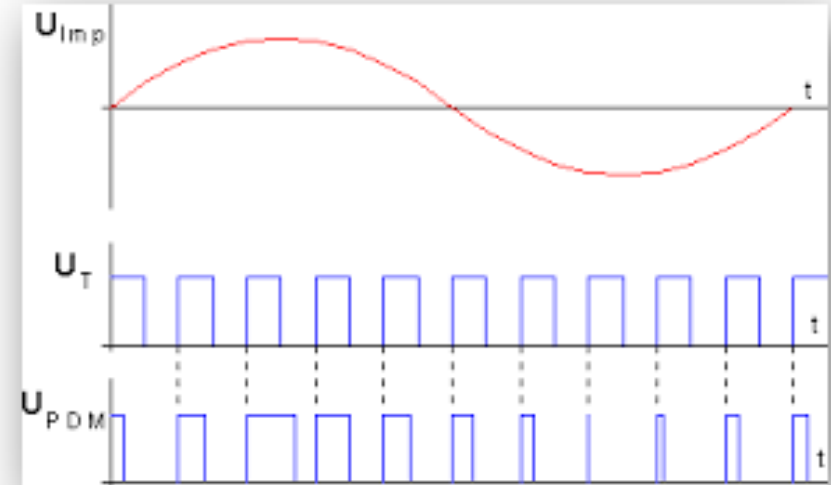
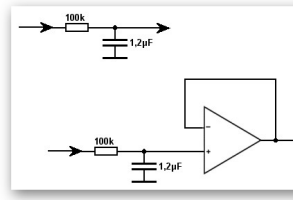
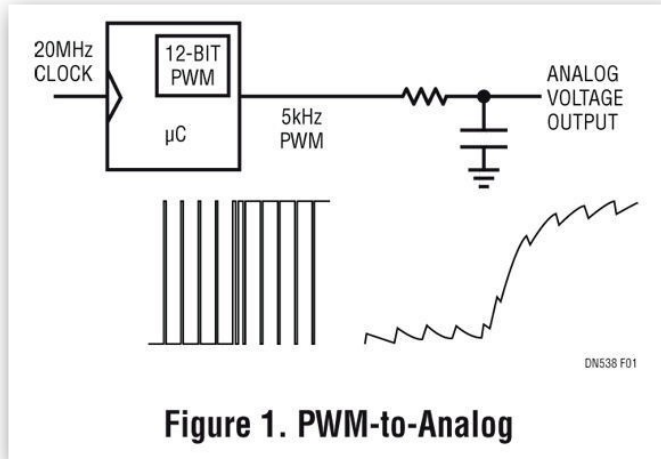
# PWM

**PWM** steht für **Puls-Weiten-Modulation**. Dabei haben die Impulse einen **konstanten Abstand (Frequenz)**, während die **Impulse unterschiedliche Länge** haben können. Damit werden z.B. Servos angesteuert oder analoge Spannungen erzeugt (DAC)



# PWM

## So funktioniert PWM als DAC



# PWM

**PWM** ist als Klasse im Modul **machine** implementiert .

Diese M5Stack Micropython Klasse ist nicht identisch mit der von Micropython.org!

Folgende Parameter sind einstellbar:

- PIN
- Frequenz
- Dutycycle

```
>>> from m5import import *  
>>> from machine import PWM  
>>> dir(machine)
```

```
>>> dir(machine.PWM)  
['__class__', '__name__', 'list',  
 '__bases__', '__dict__', 'deinit', 'duty',  
 'freq', 'hold_us', 'init', 'pause',  
 'resume']  
>>>
```

```
pwm=PWM(Pin(13), freq=300_000, duty=50)
```

# PWM

**pwm0 = PWM(10, freq=6, duty=50)** [Pin#, Hz, %]

PWM-Instanz erzeugen. PWM.init() funktioniert nicht.

**pwm0.deinit()**

PWM abschalten

**pwm0.freq([value])**

PWM-Frequenz in Hz

**pwm0.holt\_μs**

Impulslänge in μs

**pwm0.duty([value])**

Dutycycle in % in 15-bit Auflösung

Zum Testen der PWM-Funktion verwenden wir an Besten die eingebaute LED. Diese liegt auf GPIO 10.

RED LED & IR Transmitter & BUTTON A & BUTTON B		
ESP32	GPIO10	
RED LED	LED Pin	

# PWM

```
>>> from m5import import *
>>> from machine import Pin, PWM
>>> pwm0 = PWM(Pin(10))
```

```
# Als Blinker
```

```
>>> pwm0.duty(50)
>>> pwm0.freq(10)
```

```
>>> pwm0.duty(0)      # LED an
>>> pwm0.duty(100)    # LED aus
# Da LED bei 0 an ist wird der Duty-
# Cycle invertiert.
>>> pwm0.duty(20)     # DC = 100-20 = 80%
>>> pwm0.duty(80)     # DC = 100-80 = 20%
```

```
# Jetzt als Dimmer
>>> pwm0.freq(200)
>>> pwm0.duty(20)
```

Mit den Methoden **pause** und **resume** lässt sich die PWM stoppen und wieder starten.

**PWM.list()** gibt eine Liste aller definierten PWM-Instanzen aus.

# PWM

```
>>> pwm0.pause()  
# LED geht aus  
>>> pwm0.resume()  
# LED blinkt wieder
```

```
>>> PWM.list()  
PWM(10, freq=200 Hz, duty=50.00%  
[16384], duty resolution=15 bits,  
channel=0, timer=1)  
>>>
```

# PWM

## PWM-Instanz deinitialisieren und löschen:

```
>>> dir(PWM.init)
['__class__']

>>> id(pwm0)
1073496848
>>> pwm0.deinit()
>>> id(pwm0)
1073496848
>>> del(pwm0)
>>> id(pwm0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pwm0' isn't defined
>>>
```

## Wieso kommt diese Fehlermeldung?

# PWM

```
>>> from m5import import *  
from machine import Pin, PWM  
  
>>> pwm0 = machine.PWM(10, freq=5, duty=50)  
# LED blinkt  
  
>>> del(pwm0)  
# LED blinkt immer noch  
  
>>> pwm0.deinit()  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'pwm0' isn't defined  
>>>
```



So geht's richtig:

# PWM

```
>>> from m5import import *
>>> from machine import Pin, PWM
>>> pwm0 = PWM(10, freq=3, duty=50)
>>> # LED blinkt
>>> id(pwm0)
1073496704
>>> pwm0.deinit()
>>> # LED aus
>>> id(pwm0)
1073496704
>>> del(pwm0)
>>> id(pwm0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pwm0' isn't defined
>>>
```

# PWM

# ENDE