

# Zeitmessung

Zur Messung von Zeiten, z.B. die Laufzeit einer Programmsequenz, stellt Micropython **ticks** bereit. Im M5-Micropython sind **ticks** nicht vorhanden.

Bei den **ticks** handelt es sich um Werte, die im Millisekunden, Microsekunden oder CPU-Takt hochgezählt werden. Sie werden im Hintergrund vom System bearbeitet. Die **ticks** laufen vom **Systemstart** an durch. Da sie nicht unendlich groß werden dürfen, tritt irgendwann ein **Überlauf** auf. Das macht den Umgang mit **ticks** problematisch. Der maximale Wert der **ticks** ist nirgends definiert.

# Zeitmessung

Die ticks-Funktionen befinden sich im Modul **utime**.

```
>>> import utime
>>> dir(utime)
['__class__', '__name__', 'localtime',
'mktime', 'sleep', 'sleep_ms',
'sleep_us', 'ticks_add', 'ticks_cpu',
'ticks_diff', 'ticks_ms', 'ticks_us',
'time']
>>>
```

# Zeitmessung

Die `ticks_ms()`, `ticks_μs()` und `ticks_cpu()` zählen in den entsprechenden Einheiten hoch.

```
>>> from m5import import *  
>>> import utime  
  
>>> utime.ticks_ms()  
53306  
>>> utime.ticks_ms()  
71669  
>>> utime.ticks_ms()  
74877  
>>>
```

# Zeitmessung

Mit **ticks** arbeiten.

**ticks\_add(ticks, delta)**

Addiert den Wert von delta zu ticks. Ticks muss direkt aus ticks\_xx kommen.

Gemäß docs.micropython.org

```
>>> utime.ticks_add(utime.ticks_ms, 250)  
530610700
```

```
>>> utime.ticks_add(_, 250)  
530610950
```

```
>>> utime.ticks_add(250, 250)  
500
```

# Zeitmessung

Mit **ticks** arbeiten.

**ticks\_diff(ticks\_1, ticks\_2)**

```
>>> from m5import import *
>>> import utime

>>> start = utime.ticks_ms()

>>> utime.ticks_diff(start,
utime.ticks_ms())
-43493

>>>
utime.ticks_diff(utime.ticks_ms(),
start)
124085
```

# Zeitmessung

# ENDE