

Sleep

Die **Sleep-Funktionen** von Micropython erfüllen 2 Aufgaben.

1. Das Programm anzuhalten und eine bestimmte Zeit zu warten.
2. Energie zu sparen.

Diese Funktionen sind in den Modulen **machine**, **time** und **esp32** vorhanden.

Im Modul **time** sind die sleep Funktionen enthalten, die das Programm für eine vorgegebene Zeit anhalten:

- `time.sleep()`
- `time.sleep_ms()`
- `time.sleep_us()`

Diese Funktionen sind selbsterklärend.

Sleep

```
>>> import time
>>> dir(time)

['__class__', '__name__', 'localtime',
'mktime', 'sleep', 'sleep_ms',
'sleep_us', 'ticks_add', 'ticks_cpu',
'ticks_diff', 'ticks_ms', 'ticks_us',
'time']
>>>
```

Das Modul **machine** enthält die sleep Funktionen, die dem Stromsparen dienen:

- **machine.deepsleep()**
- **machine.lightsleep()**

Diesen Funktionen können Zeitangaben in ms mitgegeben werden, die die Sleepdauer begrenzen. Ausserdem können Aufweckereignisse definiert werden.

Sleep

```
>>> import machine
>>> dir(machine)
['__class__', '__name__', 'ADC', 'CAN',
'DAC', 'DEEPSLEEP', 'DEEPSLEEP_RESET',
'EXT0_WAKE', 'EXT1_WAKE', 'HARD_RESET',
'I2C', 'I2S', 'Modbus', 'ModbusSlave',
'Neopixel', 'PIN_WAKE', 'PWM',
'PWRON_RESET', 'Pin', 'RTC', 'SDCard',
'SLEEP', 'SOFT_RESET', 'SPI', 'Signal',
'TIMER_WAKE', 'TOUCHPAD_WAKE', 'Timer',
'TouchPad', 'UART', 'ULP_WAKE', 'WDT',
'WDT_RESET', 'deepsleep', 'disable_irq',
'enable_irq', 'freq', 'idle',
'lightsleep', 'mem16', 'mem32', 'mem8',
'reset', 'reset_cause', 'sleep',
'soft_reset', 'time_pulse_us',
'unique_id', 'wake_reason']
>>>
```

Sleep

Das genaue Verhalten und die Energiesparfunktionen von Lightsleep und Deepsleep hängen stark von der zugrundeliegenden Hardware ab, aber die allgemeinen Eigenschaften sind:

Lightsleep:

Der RAM-Speicher behält seinen Zustand bei. Beim Aufwachen wird die Ausführung an dem Punkt fortgesetzt, an dem **lightsleep()** angefordert wurde, wobei alle Teilsysteme betriebsbereit sind.

Deepsleep:

Es werden weder der Arbeitsspeicher noch andere Systemzustände (z. B. Peripheriegeräte oder Netzwerkschnittstellen) beibehalten. Beim Aufwachen wird die Ausführung vom Hauptskript aus wieder aufgenommen, ähnlich wie bei einem Hard- oder Power-On-Reset.

Das Modul **esp32** enthält keine weiteren Sleep Funktionen. Hier sind aber Wakeup Bedingungen definiert.

Der Stromsparaspekt von sleep wird hier nicht weiter untersucht.

Sleep

```
>>> import esp32
>>> dir(esp32)
['__class__', '__name__', 'HEAP_DATA',
'HEAP_EXEC', 'Partition', 'RMT',
'ULP', 'WAKEUP_ALL_LOW',
'WAKEUP_ANY_HIGH', 'hall_sensor',
'heap_info', 'idf_heap_info',
'nvs_erase', 'nvs_erase_all',
'nvs_getint', 'nvs_getstr',
'nvs_setint', 'nvs_setstr',
'raw_temperature', 'wake_on_ext0',
'wake_on_ext1', 'wake_on_touch']
>>>
```

Sleep

Nicht einschlafen – Es geht gleich weiter!