

Labor-Kurzleitfaden - Raspberry Pi Pico (MicroPython)

Stand: 06.10.2025

1) Voraussetzungen

- Raspberry Pi Pico / Pico W, USB-Kabel (mit Datenleitung), Steckbrett, Jumperkabel.
- PC mit VS Code + Python 3.x. VS Code Erweiterungen: „Python“ und optional „Pico-W Go“/„PyMakr“.

2) MicroPython flashen

- BOOTSEL gedrückt halten und Pico am USB anstecken → Laufwerk erscheint (RPI-RP2).
- MicroPython-UF2 auf RPI-RP2 kopieren → Pico startet neu.
- Seriellen Port in VS Code/Thonny auswählen, REPL testen (>>>).

3) Erstes Programm (LED blinken)

```
from machine import Pin
import time
led = Pin("LED", Pin.OUT)
while True:
    led.toggle()
    time.sleep(0.5)
```

4) Taster mit Pull-Up

- Interner Pull-Up: Pin als Eingang mit Pull-Up konfigurieren; Taster nach GND.

```
from machine import Pin
btn = Pin(15, Pin.IN, Pin.PULL_UP)
print(btn.value()) # 1 = offen, 0 = gedrückt
```

5) Servo (optional)

- Extern mit 5V versorgen; GND mit Pico verbinden; Signal über PWM-Pin (z. B. GP16).

```
from machine import Pin, PWM
pwm = PWM(Pin(16)); pwm.freq(50)
# 1ms..2ms Puls je nach Servo
pwm.duty_u16(1638) # ~1ms (Min)
# pwm.duty_u16(8192) # ~1.5ms (Mitte)
# pwm.duty_u16(14746) # ~2ms (Max)
```

6) Hinweise

- Niemals 5V direkt an Pico-GPIO! HC-SR04 Echo auf 3.3V teilen. GNDs immer verbinden.
- Fehlersuche: Kabel prüfen, richtigen Port wählen, UF2 neu flashen, Baudrate 115200.