

M5 - GPIO

Pins und GPIO aus der MP-Doku

Verwenden Sie die Klasse **machine.Pin**.

Verfügbare Pins sind aus den folgenden Bereichen (einschließlich): **0-19, 21-23, 25-27, 32-39**.

Diese entsprechen den tatsächlichen GPIO-Pin-Nummern des ESP32-Chips. Beachten Sie, dass viele Endbenutzer-Boards ihre eigene Adhoc-Pinnummerierung verwenden (z.B. D0, D1, ...).

Für die Zuordnung zwischen den logischen Pins der Platine und den physischen Pins des Chips konsultieren Sie die Dokumentation Ihrer Platine.

M5 - GPIO

- **Pins 1 und 3** sind **REPL UART TX** bzw. **RX**
- Die **Pins 6, 7, 8, 11, 16** und **17** werden für den Anschluss des **eingebetteten Flash** verwendet und sind für andere Zwecke nicht geeignet.
- Die **Pins 34-39** sind **nur Eingänge** und haben auch keine internen Pull-up-Widerstände.
- Der Pull-Wert einiger Pins kann auf **Pin.PULL_HOLD** gesetzt werden, um den **Stromverbrauch** während des Deepsleep zu **reduzieren**.

M5 - GPIO

```
from machine import Pin
```

```
p0 = Pin(0, Pin.OUT)      # create output pin on GPIO0
p0.on()                   # set pin to "on" (high) level
p0.off()                  # set pin to "off" (low) level
p0.value(1)               # set pin to on/high
```

```
p2 = Pin(2, Pin.IN)       # create input pin on GPIO2
print(p2.value())         # get value, 0 or 1
```

```
p4 = Pin(4, Pin.IN, Pin.PULL_UP) # enable internal pull-up resistor
p5 = Pin(5, Pin.OUT, value=1)    # set pin high on creation
```

M5 - GPIO

Die PIN Ansteuerung der UIFlow-IDE

```
import machine

pin0 = machine.Pin(0, mode=machine.Pin.IN, pull=machine.Pin.PULL_UP)
pin0.on()
pin0.off()
pin0.value(0)
temp = pin0.value()

from easyIO import *
digitalWrite(26, 0)
toggleIO(26)
temp = digitalRead(0)
temp = map_value(0, 0, 1023, 0, 4) # Wert von einem Bereich auf einen anderen
übertragen. Wert nach oben oder unten skalieren
```

M5 - GPIO

Die Methoden der Klasse `machine.Pin`

```
>>> import machine
```

```
>>> dir(machine.Pin)
```

```
['__class__', '__name__', 'value', '__bases__', '__dict__', 'IN',  
'IRQ_FALLING', 'IRQ_RISING', 'OPEN_DRAIN', 'OUT', 'PULL_DOWN', 'PULL_HOLD',  
'PULL_UP', 'WAKE_HIGH', 'WAKE_LOW', 'init', 'irq', 'off', 'on']
```

```
>>>
```

```
# Von https://github.com/m5stack/UIFlow-Code/wiki/Advanced
```

```
# mode: machine.Pin.(IN, OUT, OUT_OD, INOUT, OPEN_DRAIN)
```

```
# pull: machine.Pin.(PULL_UP, PULL_DOWN, PULL_HOLD, PULL_FLOAT) # Error
```

```
# value: 0, 1
```

M5 - GPIO

ENDE