

TxPi-HAT Python Schnellstart

Peter Habermehl, MINTronics

info@mintronics.de



Mit dieser Anleitung möchten wir den schnellen Einstieg in die Python-Programmierung des TxPi-HAT ermöglichen.

Dazu stellen wir ein Python-Modul „TxPiHAT.py“ zur Verfügung, das das Ansprechen der beiden analogen Ausgänge sowie der vier digitalen Eingänge auf sehr einfache Weise ermöglicht.

Zu finden ist das Modul unter www.mintronics.de auf der Produkt-Seite des TxPi-HAT: <https://www.mintronics.de/ft-hat> unter „Weitere Informationen / Python Schnellstart“.

Nach dem Download ist das Modul „txpihat.py“ im Python-Suchpfad abzulegen. Dies kann auch das Verzeichnis sein, in dem eigene Programme erstellt werden.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from txpihat import * # Modul laden

from time import sleep # Delay-Funktion aus dem time-Modul laden

# create HAT object
try:
    hat = TxPiHAT()
except Exception as e: # falls das nicht geklappt hat....
    print(str(e))
    exit()

# Die Ausgänge heißen "M1" und "M2"
# Mögliche Status sind "Off", "Right", "Left", "Brake"
# Sie werden mit m_set_mode() gesetzt:

hat.m_set_mode("M1", "Brake") # Motor 1 gebremst aus
hat.m_set_mode("M2", "Right") # Motor 2 rechts

# Der PWM-Wert für einen Ausgang kann 0 - 100 betragen
# Er wird mit m_set_pwm() gesetzt:

hat.m_set_pwm("M1", 0) # Motor 1 auf 0%
hat.m_set_pwm("M2", 75) # Motor 2 auf 75%

sleep(5) # 5 Sekunden warten

# Der gelesene Status eines Einganges kann
# True oder False sein. Die Eingänge heißen "I1" - "I4"
# Gelesen wird der Eingang mit get_input():

for i in ["I1", "I2", "I3", "I4"]:
    print("Eingang " + i + " hat den Status: ", hat.get_input(i))
```

Informationen zur Nutzung der Echtzeit-Uhr und des I²C-Busses finden sich im Handbuch zum HAT, das ebenfalls über unsere Webseite verfügbar ist.