

Motorsteuerung

Speichere diesen Code auf dem Pico unter dem Namen "motor.py".

Motorbibliothek

```
from machine import Pin, PWM
import utime

MIN_DUTY = 0
MAX_DUTY = 60000
MAX_SPEED = 100
MIN_SPEED = 30

class Motor:

    """This class manages the motor. Don't edit!"""

    def __init__(self, pinNo):

        self.gpio = pinNo

        self.speed=0

        self.forward=True

        self.pwm1=PWM(Pin(pinNo))

        self.pwm1.freq(50000)

        self.pwm1.duty_u16(0)

        self.pwm2=PWM(Pin(pinNo+1))

        self.pwm2.freq(50000)

        self.pwm2.duty_u16(0)

        self.speed_offset = 0


    def set_speed(self,s):

        """Sets the speed of the motor. Checks for sensible input."""

        if s + self.speed_offset <= MIN_SPEED:

            s = 0

            self.reset_offset()

        elif s + self.speed_offset >= MAX_SPEED:

            s = MAX_SPEED

        self.pwm1.duty_u16(int(MAX_DUTY*(s+self.speed_offset)/100))
```

```

self.speed=s

def change_speed(self,sc):
    """This defines an offset to the speed in motor. It is used with the remote control to turn the robot."""
    if self.speed + sc > MIN_SPEED and self.speed + sc < MAX_SPEED:
        self.speed_offset += sc
        self.set_speed(self.speed)

def reset_offset(self):
    self.speed_offset = 0

def off(self):
    self.pwm1.duty_u16(0)
    self.speed = 0

def set_forward(self,forward):
    """Sets the motor to forward or backward without changing the speed. """
    if self.forward==forward:
        return
    self.pwm1.duty_u16(0)
    self.pwm1,self.pwm2=self.pwm2,self.pwm1
    self.forward=forward
    self.set_speed(self.speed)
    #self.pwm1.duty_u16(int(MAX_DUTY*(self.speed+self.speed_offset)/100)) # uncommenting this causes
problems with the remote control. After changing
    # the direction the robot would drive even if the remote control speed said 0.

```

Beispiel für die Anwendung dieser Bibliothek

Kopiere diesen Code in eine andere Datei auf dem Pico, z. B. „motortest.py“.

```

from motor import Motor
from utime import sleep, sleep_ms

motor = Motor(12)

```

```
motor.set_speed(70)
motor.set_forward(True)
sleep(1)
motor.off()
```

Revision #11

Created 31 August 2022 15:12:16 by Marcus Jacobs

Updated 24 September 2024 06:52:51 by Marcus Jacobs