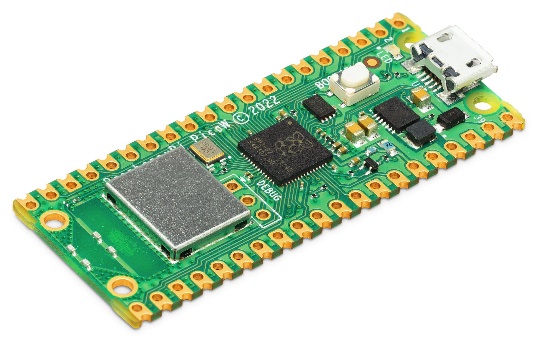
**Temperatur per Mail senden**

Dieses Skript sendet die gemessene Temperatur des Pico per Mail an den festgelegten Empfänger. Erweiterung DHT22



PICO W [DHT22](https://www.roboter-bausatz.de/p/dht22-digitaler-temperatur-und-feuchtigkeitssensor)

import umail

import network

import time

import machine

import sys

import time

import usocket as socket

import ustruct as struct

from time import sleep

from dht import DHT22

from machine import Pin

# Your email details bei [Google](http://www.luowice.com/de/soft/y4/Einrichten%20der%20Bewegungserkennung%20und%20E-Mail-Benachrichtigung-Y4.pdf) anlegen

sender\_email = 'maxmustermann@gmail.com‘

sender\_name = 'RaspberryPiPico'

sender\_app\_password = 'passowrt von Google'

recipient\_email =Empfänger Mail

email\_subject ='Email from RPi Pico'

# Your network credentials

ssid = 'Wlan Name'

password = 'Wlan Passwort'

# Initialisierung GPIO und DHT22

sleep(1)

dht22\_sensor = DHT22(Pin(27, Pin.IN, Pin.PULL\_UP))

# NTP-Host

NTP\_HOST = 'pool.ntp.org'

# Winterzeit / Sommerzeit

GMT\_OFFSET = 3600 \* 1 # 3600 = 1 h (Winterzeit)

#GMT\_OFFSET = 3600 \* 2 # 3600 = 1 h (Sommerzeit)

#Connect to Wi-Fi

wlan = network.WLAN(network.STA\_IF)

wlan.active(True)

wlan.connect(ssid, password)

# Wait for connection to establish

max\_wait = 10

while max\_wait > 0:

if wlan.status() < 0 or wlan.status() >= 3:

break

max\_wait -= 1

print('waiting for connection...')

time.sleep(1)

# Manage connection errors

if wlan.status() != 3:

raise RuntimeError('Network Connection has failed')

else:

print('connected')

# Funktion: Zeit per NTP holen

def getTimeNTP():

NTP\_DELTA = 2208988800

NTP\_QUERY = bytearray(48)

NTP\_QUERY[0] = 0x1B

addr = socket.getaddrinfo(NTP\_HOST, 123)[0][-1]

s = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

try:

s.settimeout(1)

res = s.sendto(NTP\_QUERY, addr)

msg = s.recv(48)

finally:

s.close()

ntp\_time = struct.unpack("!I", msg[40:44])[0]

return time.gmtime(ntp\_time - NTP\_DELTA + GMT\_OFFSET)

# Funktion: RTC-Zeit setzen

def setTimeRTC():

# NTP-Zeit holen

tm = getTimeNTP()

machine.RTC().datetime((tm[0], tm[1], tm[2], tm[6], tm[3], tm[4], tm[5], 0))

# Zeit setzen

setTimeRTC()

datetime = machine.RTC().datetime()

mailtext=str('Aktuelle Uhrzeit: %02d:%02d' % (datetime[4], datetime[5]))

# Send email once after MCU boots up

def mailersend():

smtp = umail.SMTP('smtp.gmail.com', 465, ssl=True)

smtp.login(sender\_email, sender\_app\_password)

smtp.to(recipient\_email)

smtp.write("From:" + sender\_name + "<"+ sender\_email+">\n")

smtp.write("Subject:" + email\_subject + "\n")

smtp.write(mailtext)

smtp.send()

smtp.quit()

i = 1

while i < 2:

print(i)

print(mailtext)

mailersend()

sleep(2)

setTimeRTC()

datetime = machine.RTC().datetime()

mailtext=str('Aktuelle Uhrzeit: %02d:%02d' % (datetime[4], datetime[5]))

i += 1

#datetime = machine.RTC().datetime()

print('Aktuelle Uhrzeit: %02d:%02d' % (datetime[4], datetime[5]))

while True:

# Messung durchführen

dht22\_sensor.measure()

# Werte lesen

temp = dht22\_sensor.temperature()

humi = dht22\_sensor.humidity()

# Werte ausgeben

print(' Temperatur:', temp, '°C')

print('Luftfeuchtigkeit:', humi, '%')

# + str(temp) + ' Feuchte : ' + str(humi) + ' ' + 'Aktuelle Uhrzeit: ' + str(datetime[4]) + ':' + str(datetime[5])

mailtext=str(temp)

print(mailtext)

if temp > 20:

mailersend()

else:

mailtext=str('Aktuelle Uhrzeit: %02d:%02d' % (datetime[4], datetime[5]))

mailersend()

sleep(600)