

Irgendwas mit MicroController

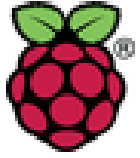
Jochen Krapf

KNF-Kongress 2022

Agenda

Irgendwas...

verfügbar?



Dieser Artikel steht derzeit nicht zur Verfügung!

Benachrichtigen Sie mich, sobald der Artikel lieferbar ist.

Ihre E-Mail-Adresse



Welcher MicroController?

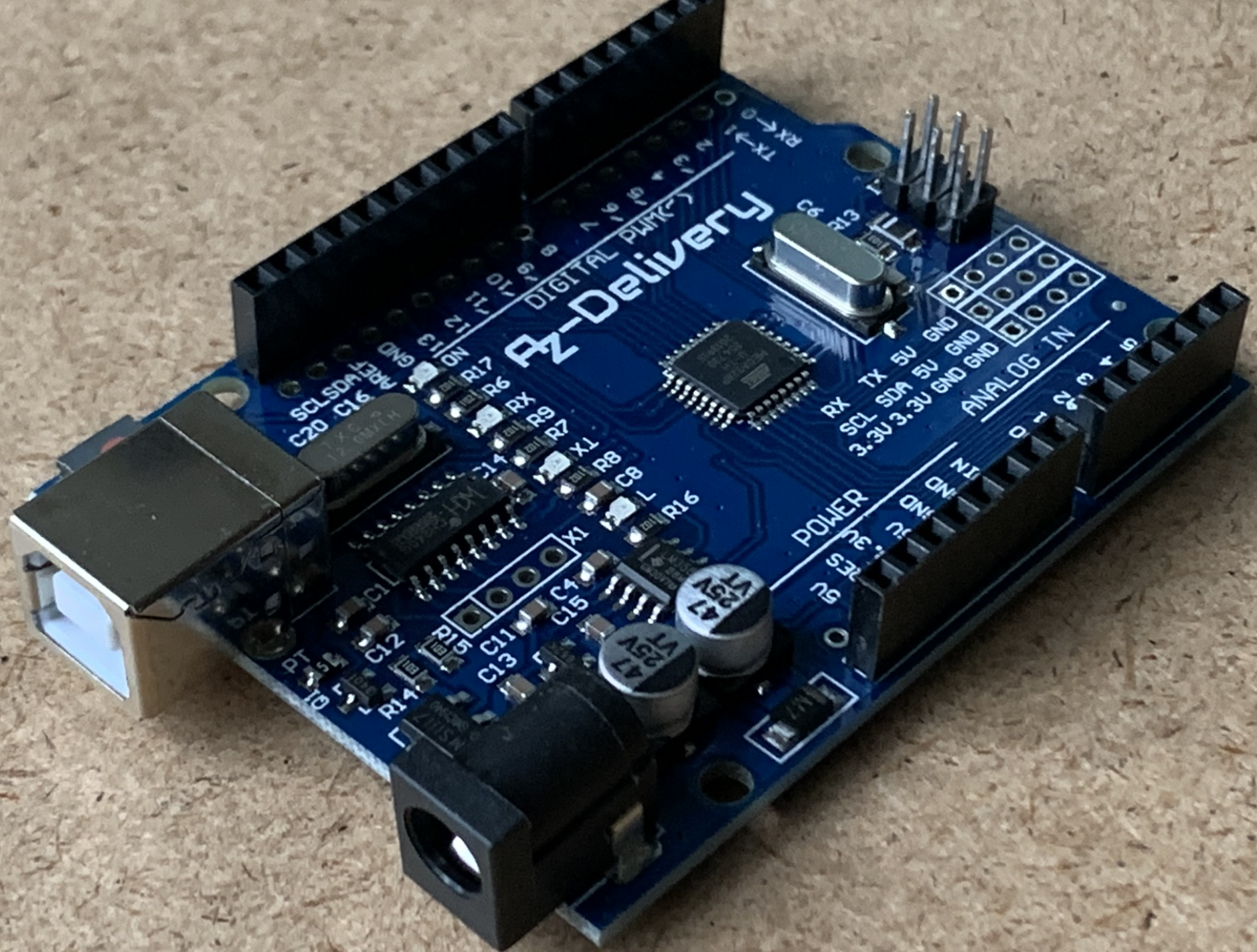
µC:

- Arduino (AVR ATmega)
- STM32
- ESP8266
- ESP32
 - M5Stack-Familie
- Raspberry Pi Pico (W)
- Raspberry Pi 1, 2, 3, 4, Zero (2)(W)
- '*' Pi

Welcher MicroController?

Arduino (AVR ATmega)

- *teilweise verfügbar*
- 8-Bit
- sehr wenig Speicher, langsam
- I2C, SPI, ADC,
- große Community



Welcher MicroController?

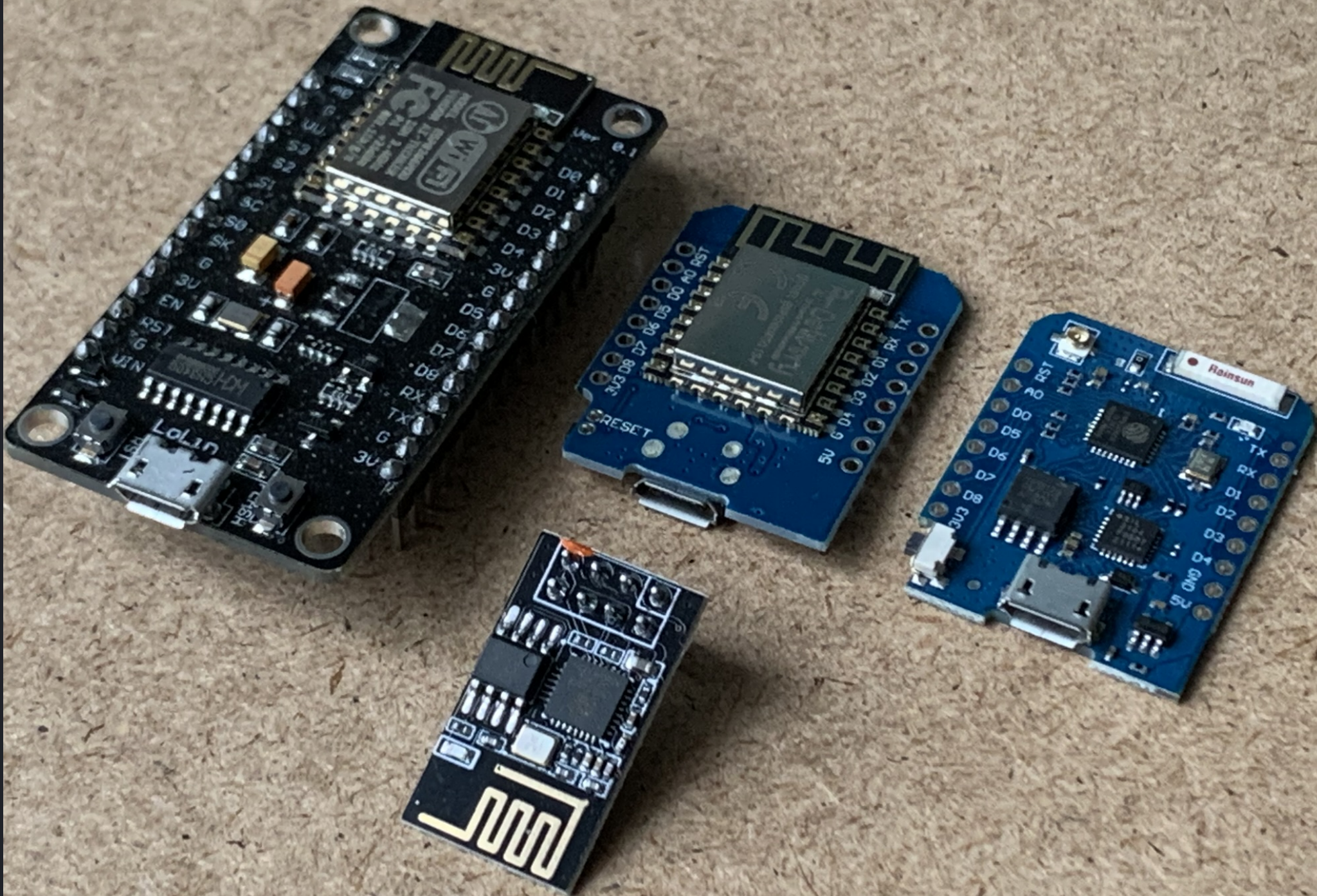
STM32

- *gar nicht verfügbar*
- 32-Bit, (FPU)
- wenig Speicher, schnell
- I2C, SPI, ADC, (CAN), ...
- Industrie
- C/C++

Welcher MicroController?

ESP8266

- *verfügbar*
- 32-Bit
- genug Speicher, schnell
- WLAN, I2C, SPI, ADC
- große Community
- C/C++, MicroPython



Welcher MicroController?

ESP32

- *verfügbar*
- 32-Bit
- viel Speicher, sehr schnell
- WLAN, BT, I2C, SPI, ADC, DAC, Touch, Hall
- große Community
- C/C++, MicroPython





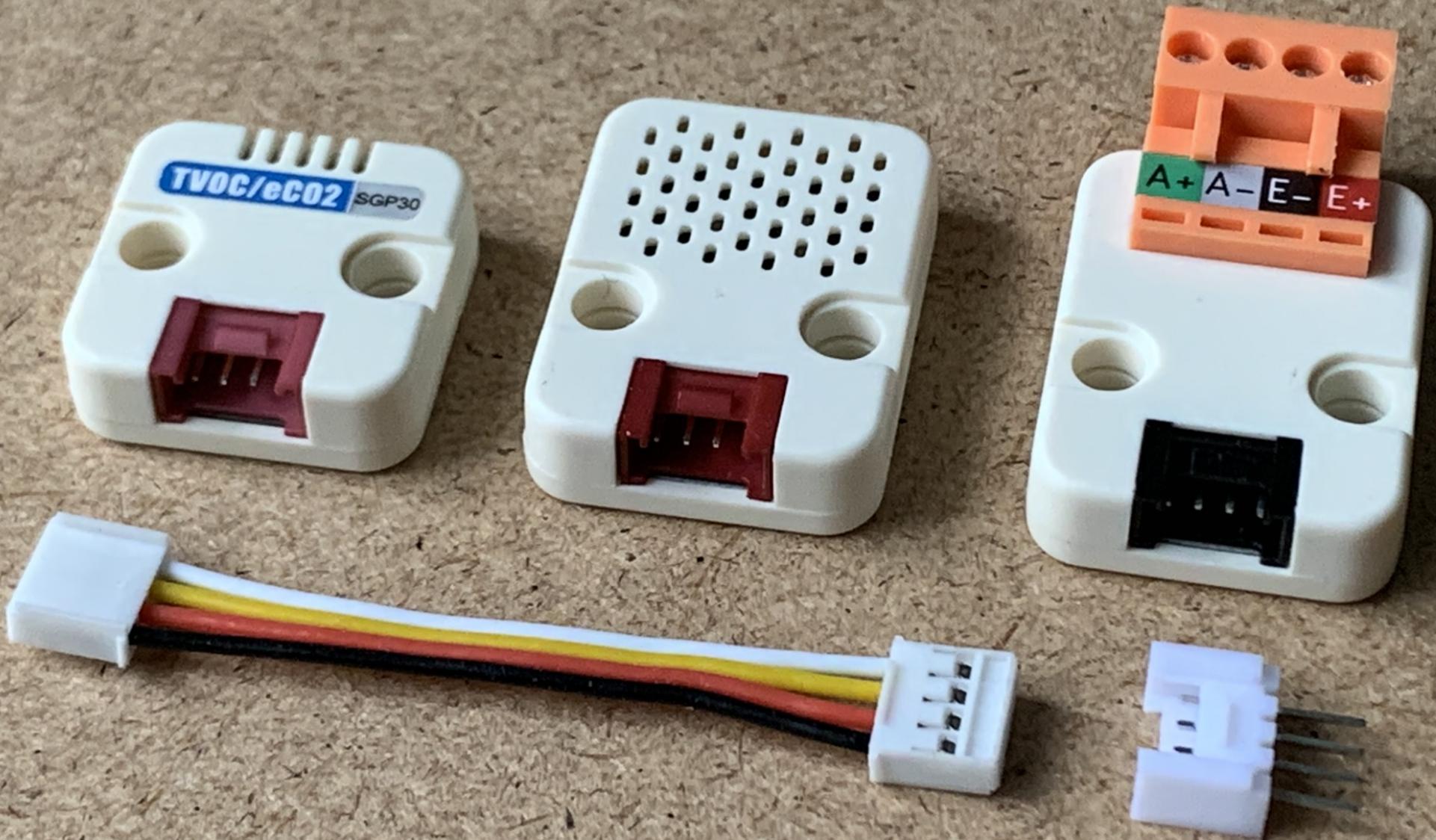
Welcher MicroController?

M5Stack-Familie (ESP32)

- *teilweise verfügbar*
- Intern: ESP32
- viel Doku
- viele Sensoren und Aktoren
- Blockly, MicroPython



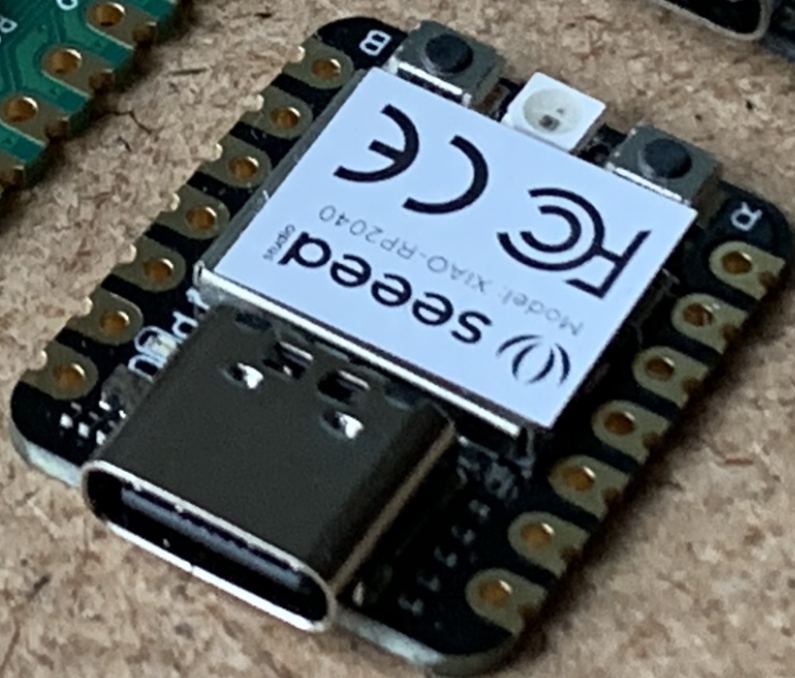
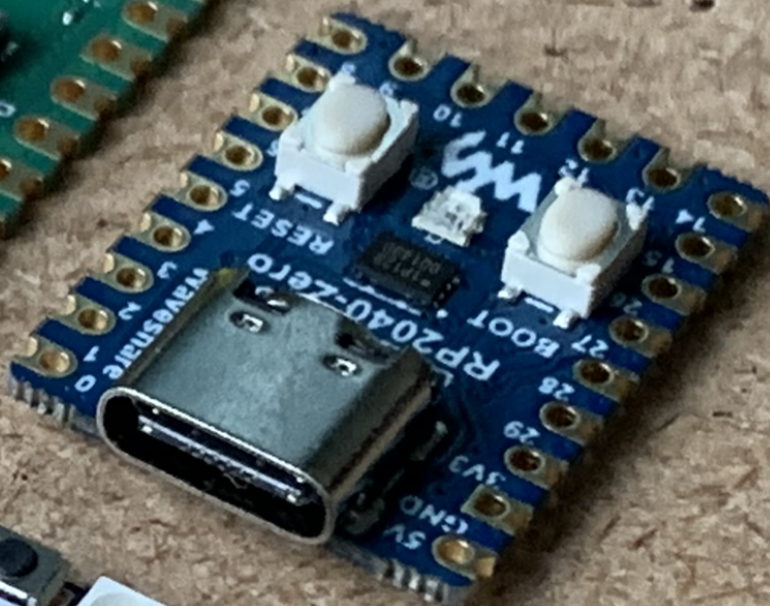
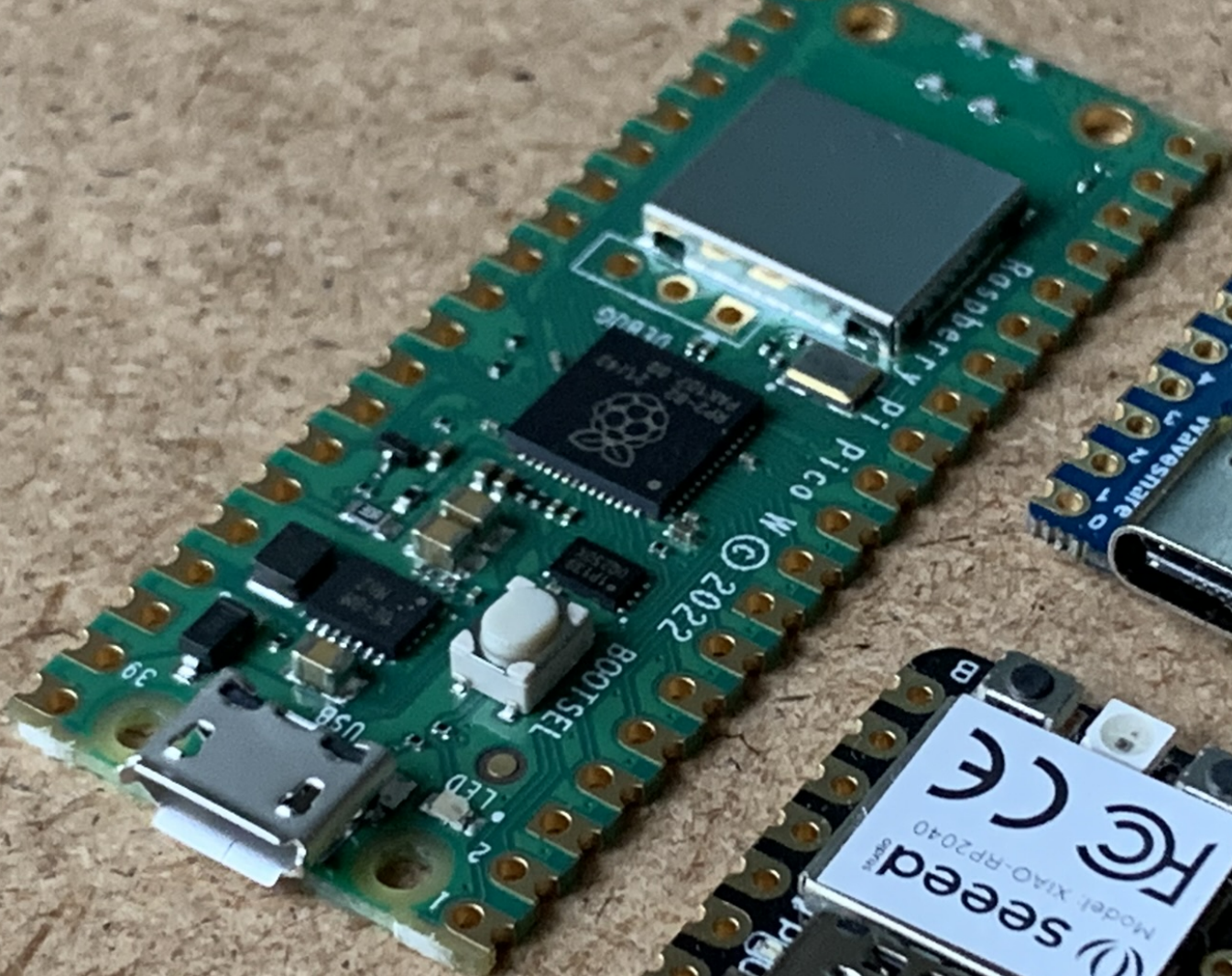




Welcher MicroController?

Raspberry Pi PICO (W)

- *gut verfügbar*
- Prozessor RP2040
- 32-Bit
- viel Speicher, sehr schnell
- (WLAN), I2C, SPI, ADC
- große Community
- PIO (programmatische Port-Bausteine)
- C/C++, MicroPython



Welcher MicroController?

Raspberry Pi 1, 2, 3, 4, Zero (2)(W)

- *nicht verfügbar / teuer*
- Prozessor Broadcom
- 32/64-Bit
- unendlich Speicher, extrem schnell
- (WLAN), (Ethernet), USB, I2C, SPI
- große Community, veraltete/unpassende Infos
- Linux (Vorteil/Nachteil)
- C/C++, CPython, ...



Welcher MicroController?

'*' Pi

Orange Pi, Banana Pi, Odroid, ...

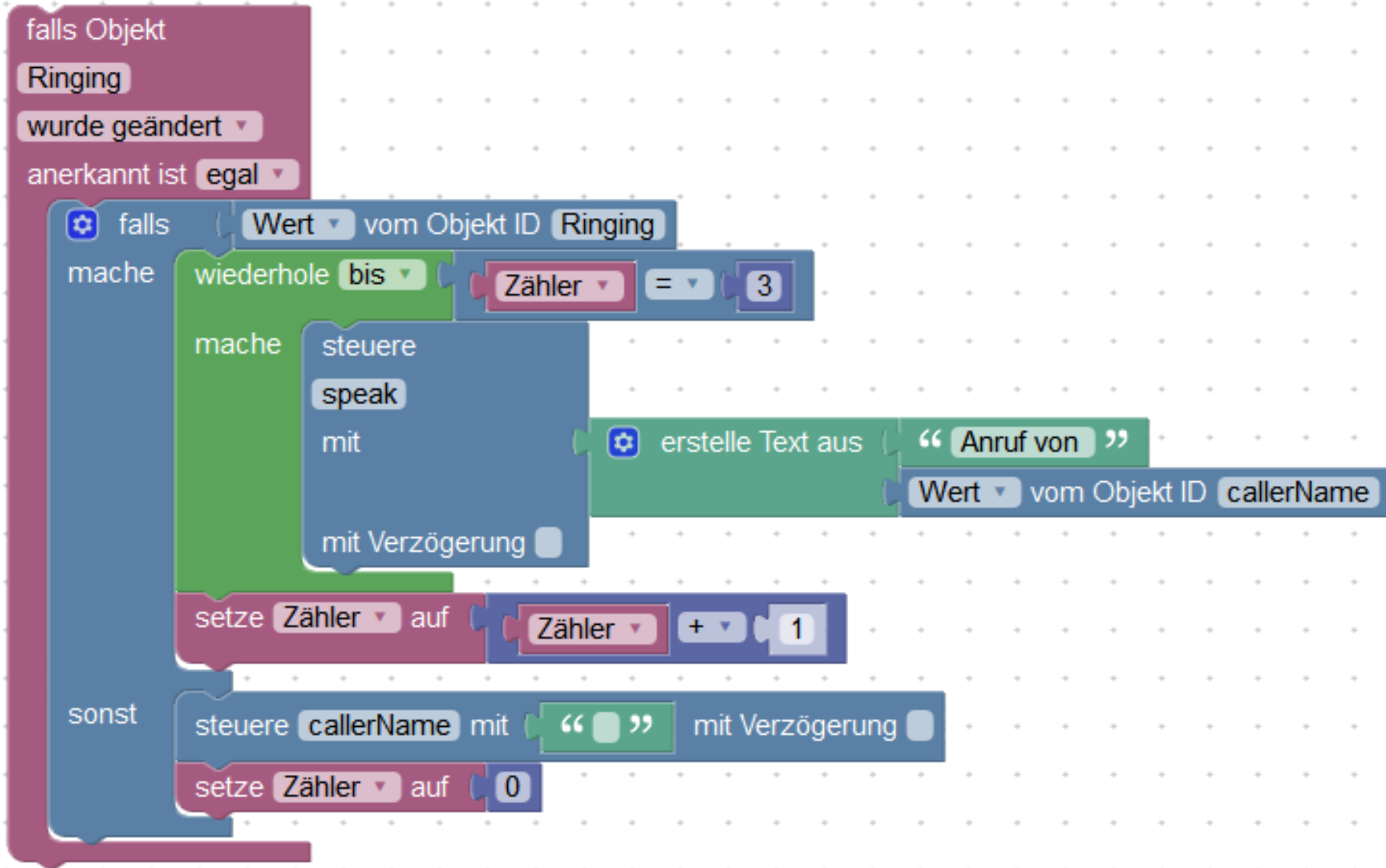
- *verfügbar / etwas teurerer*
- Prozessor Allwinner H2, H3, H5, A20, ...
- 32/64-Bit
- unendlich Speicher, extrem schnell
- (WLAN), (Ethernet), USB, I2C, SPI
- teilweise inkompatibel zu RPi
- C/C++, CPython, ...

Welcher Programmiersprache?

Sprachen:

- ~~Assembler~~
- C/C++
- MicroPython
 - CircuitPython
- Rust?
- Blockly (für M5Stack)

Blockly



Welche IDE?

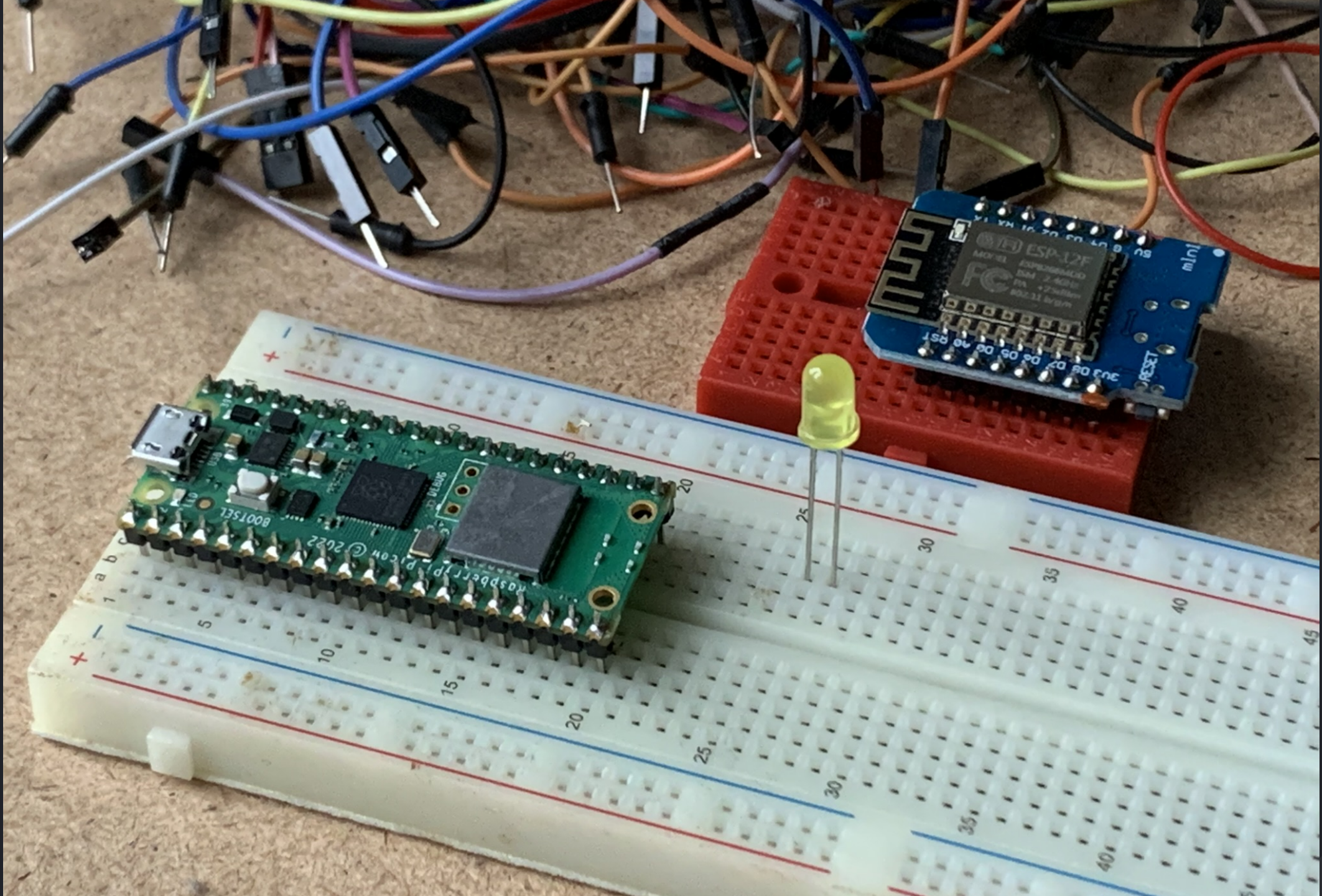
IDEs:

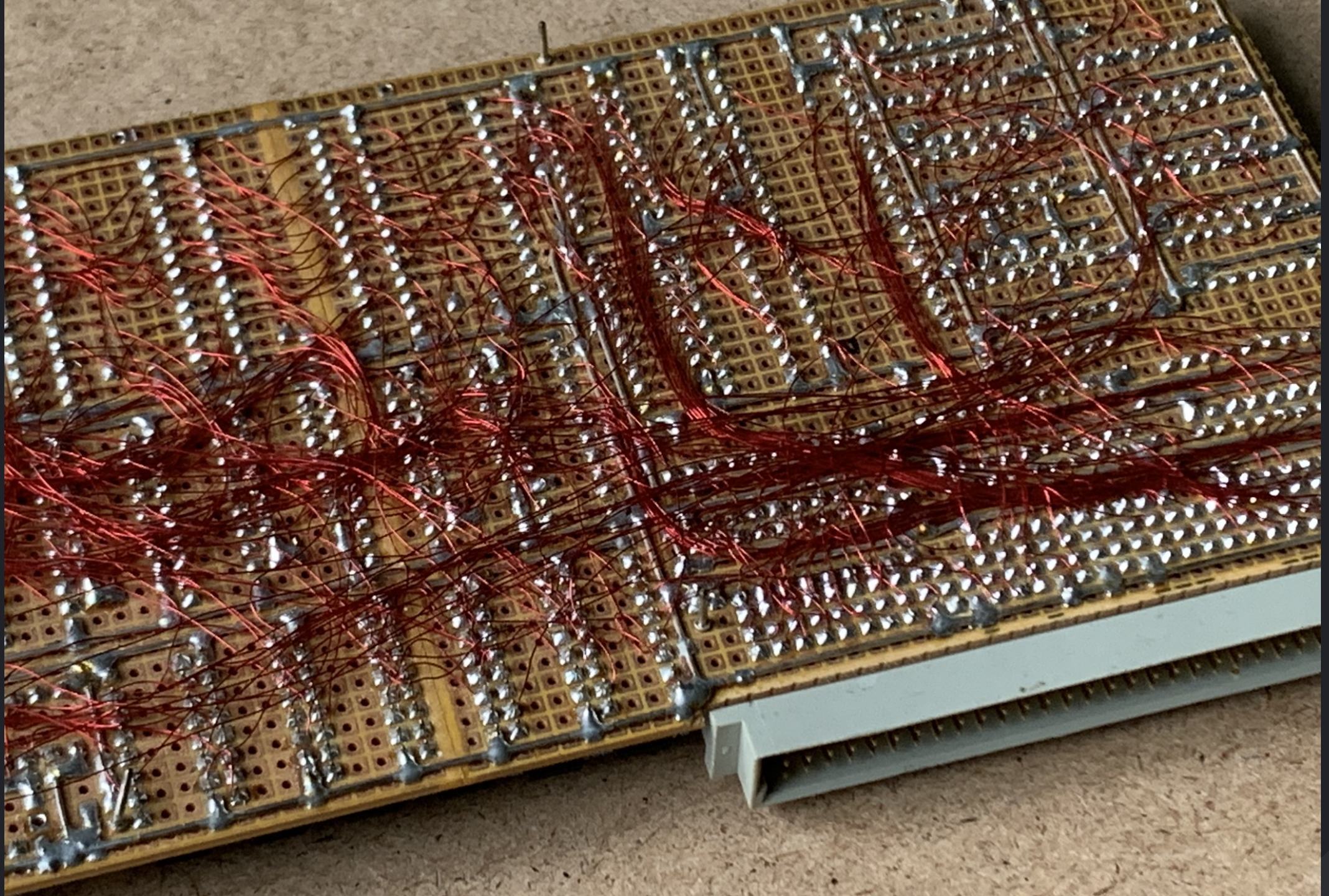
- Arduino IDE für Arduino, ESP* (C/C++)
- ~~uPyCraft für ESP* (MicroPython)~~
- Thonny für ESP*, Pico (MicroPython)
- VisualStudio Code (alle)

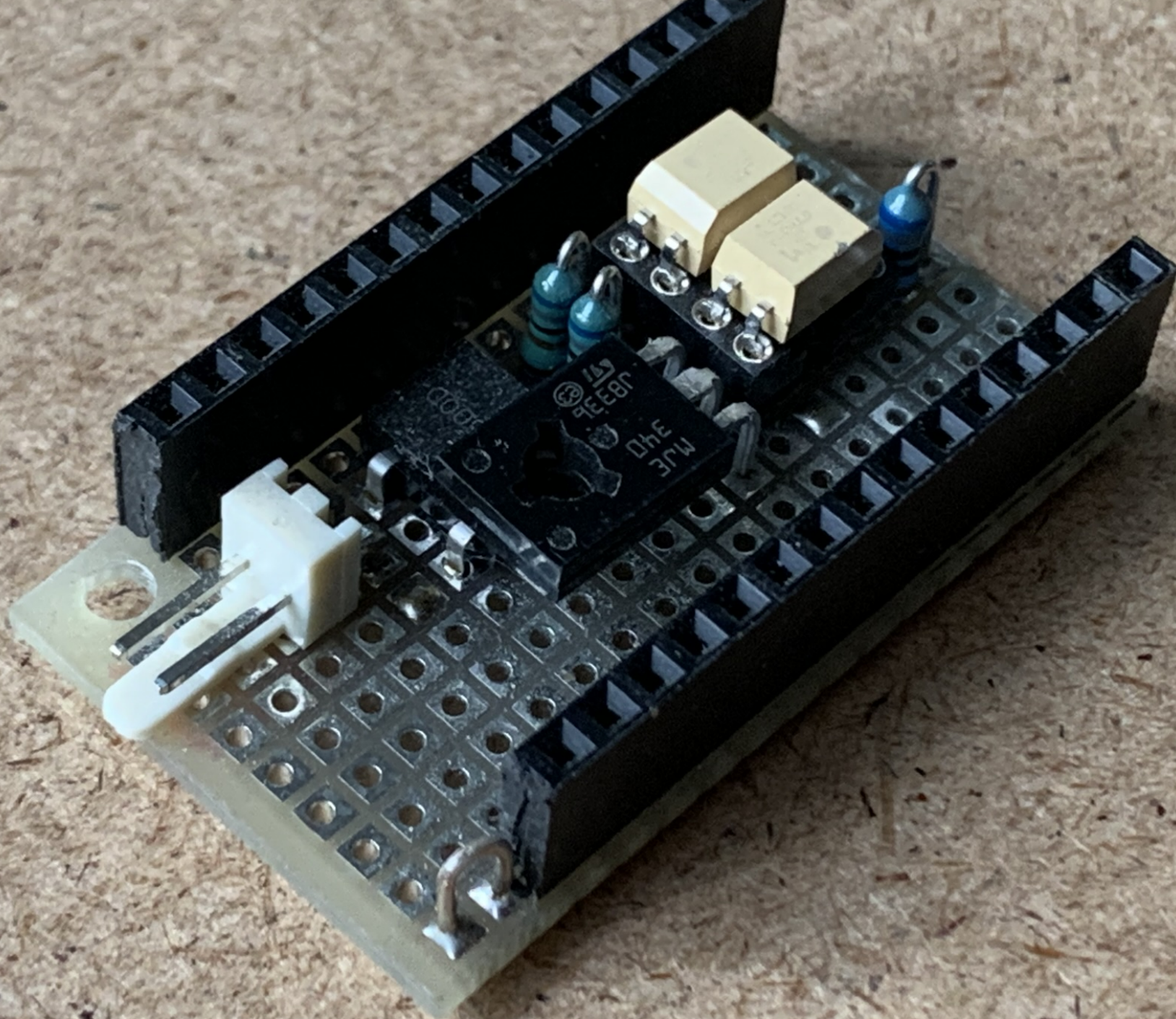
Löten? Stecken?

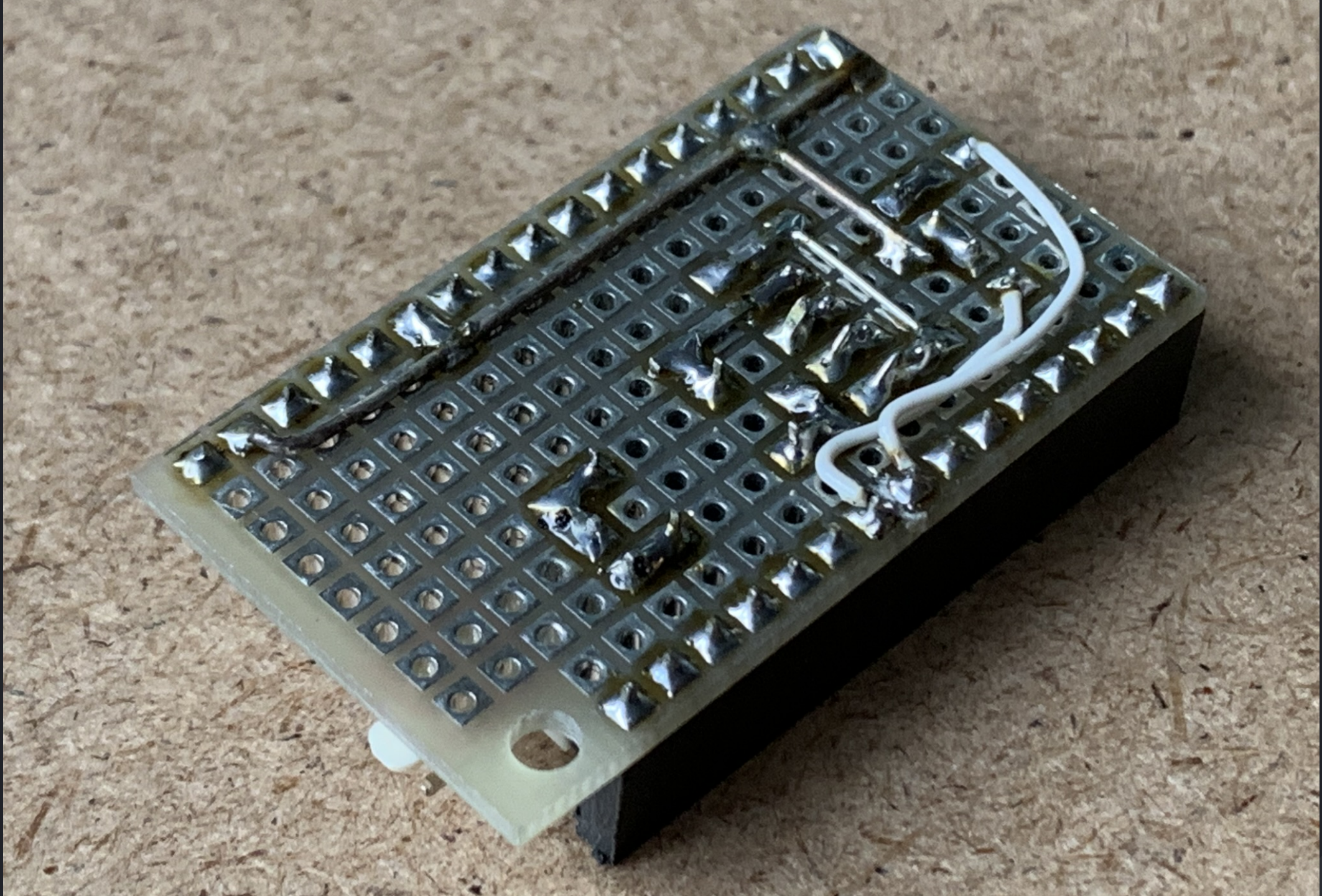
Methoden:

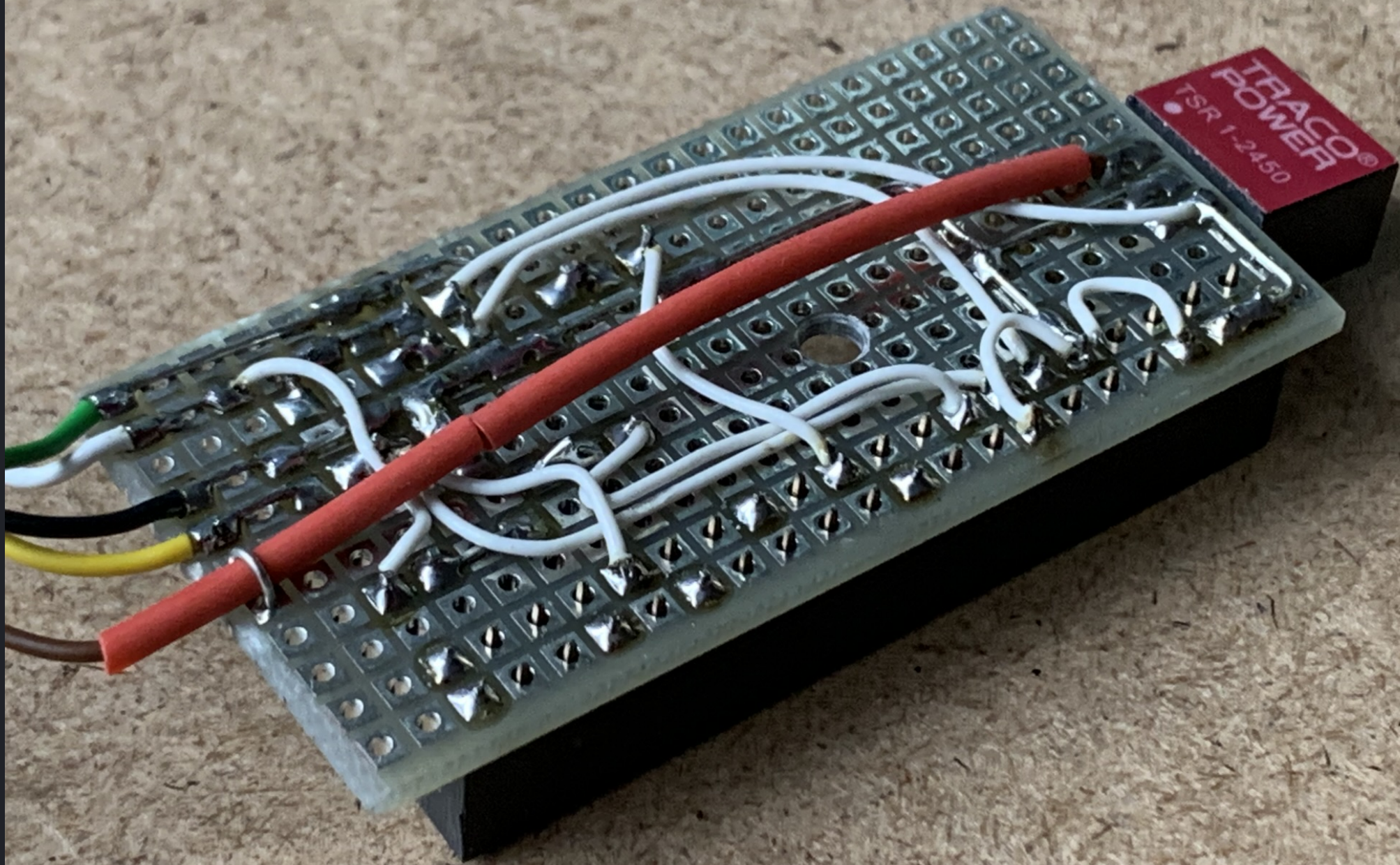
- Breadboard
- Lochraster / Fädeln
- ~~PCB selbst ätzen~~
- PCB aus China

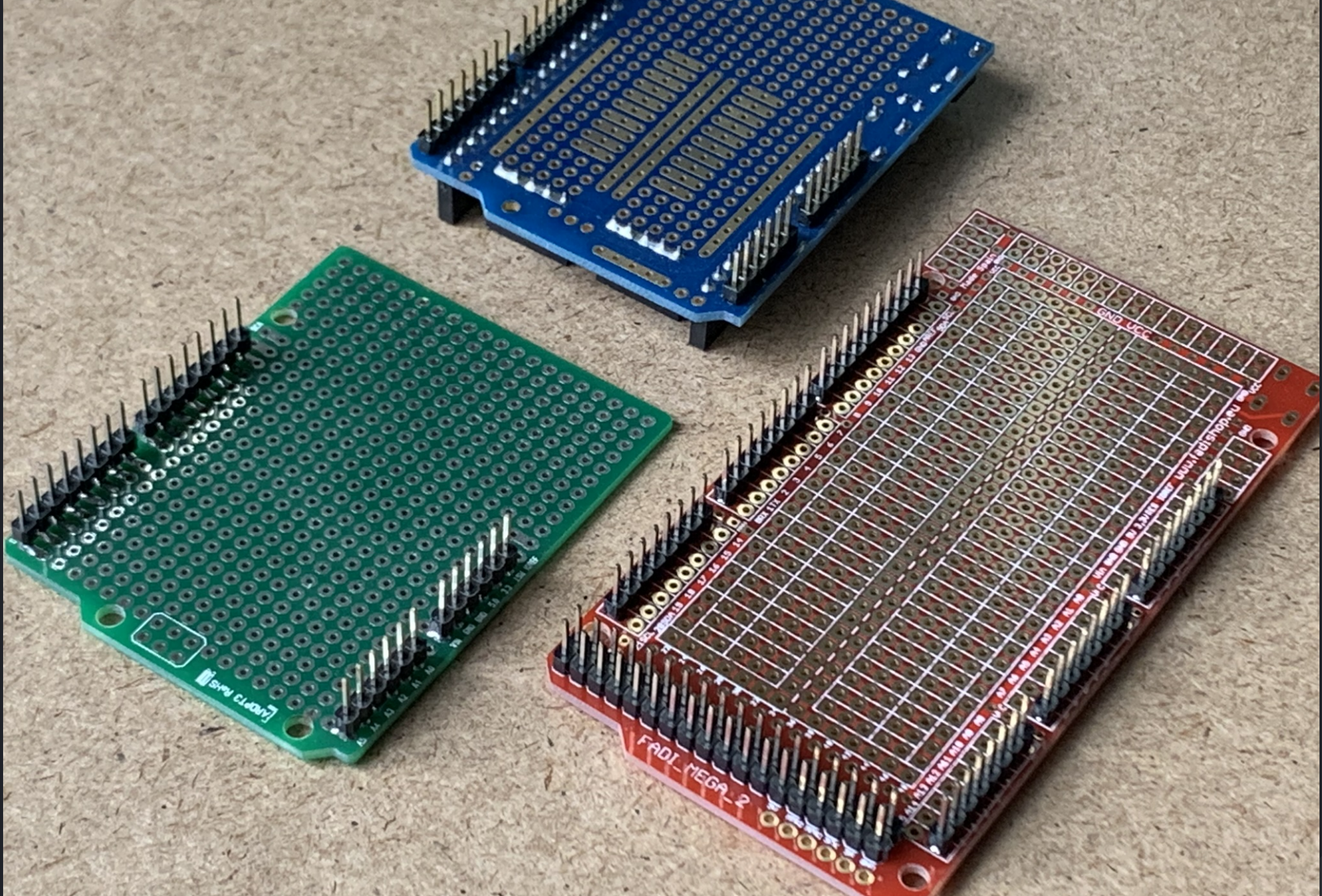


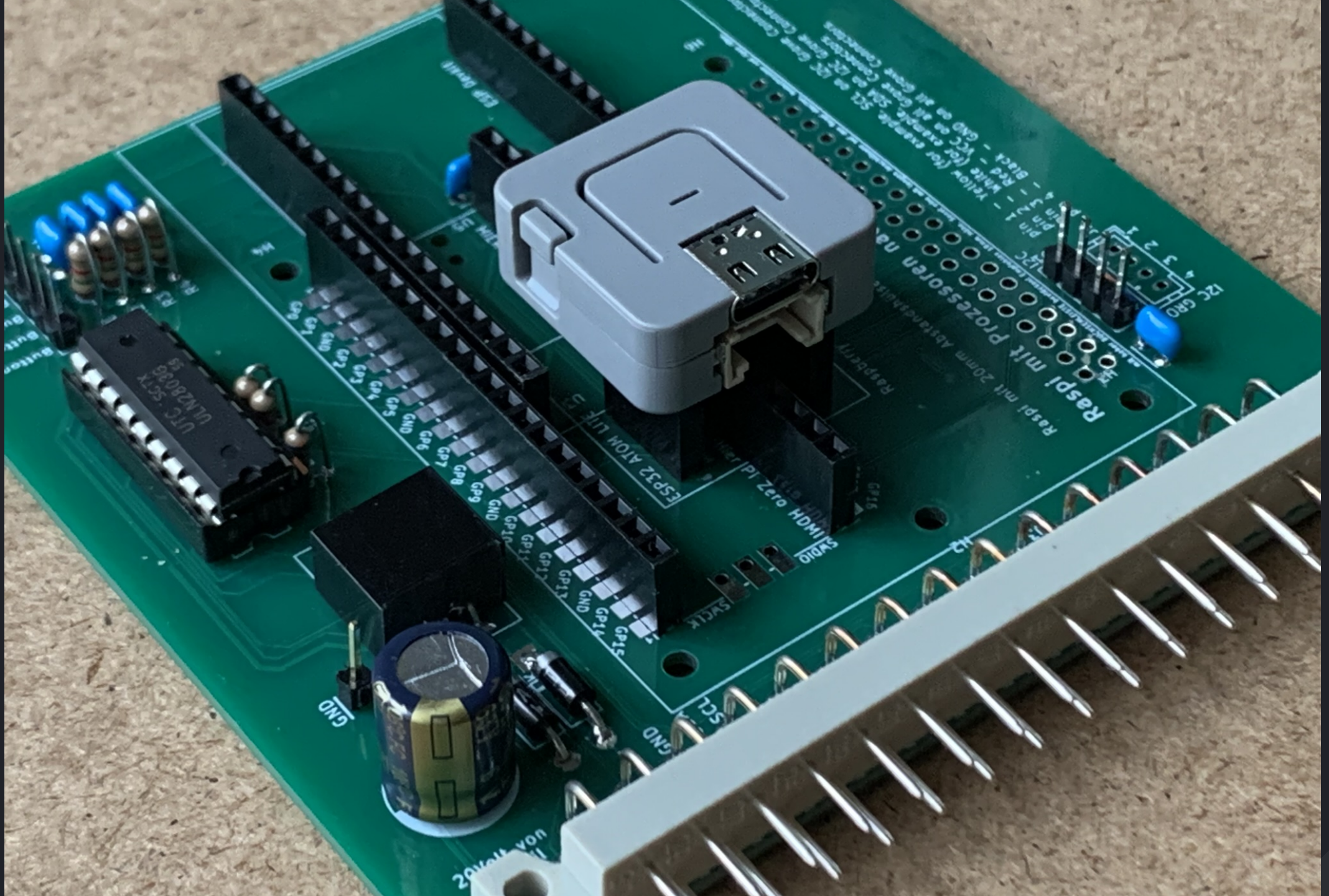


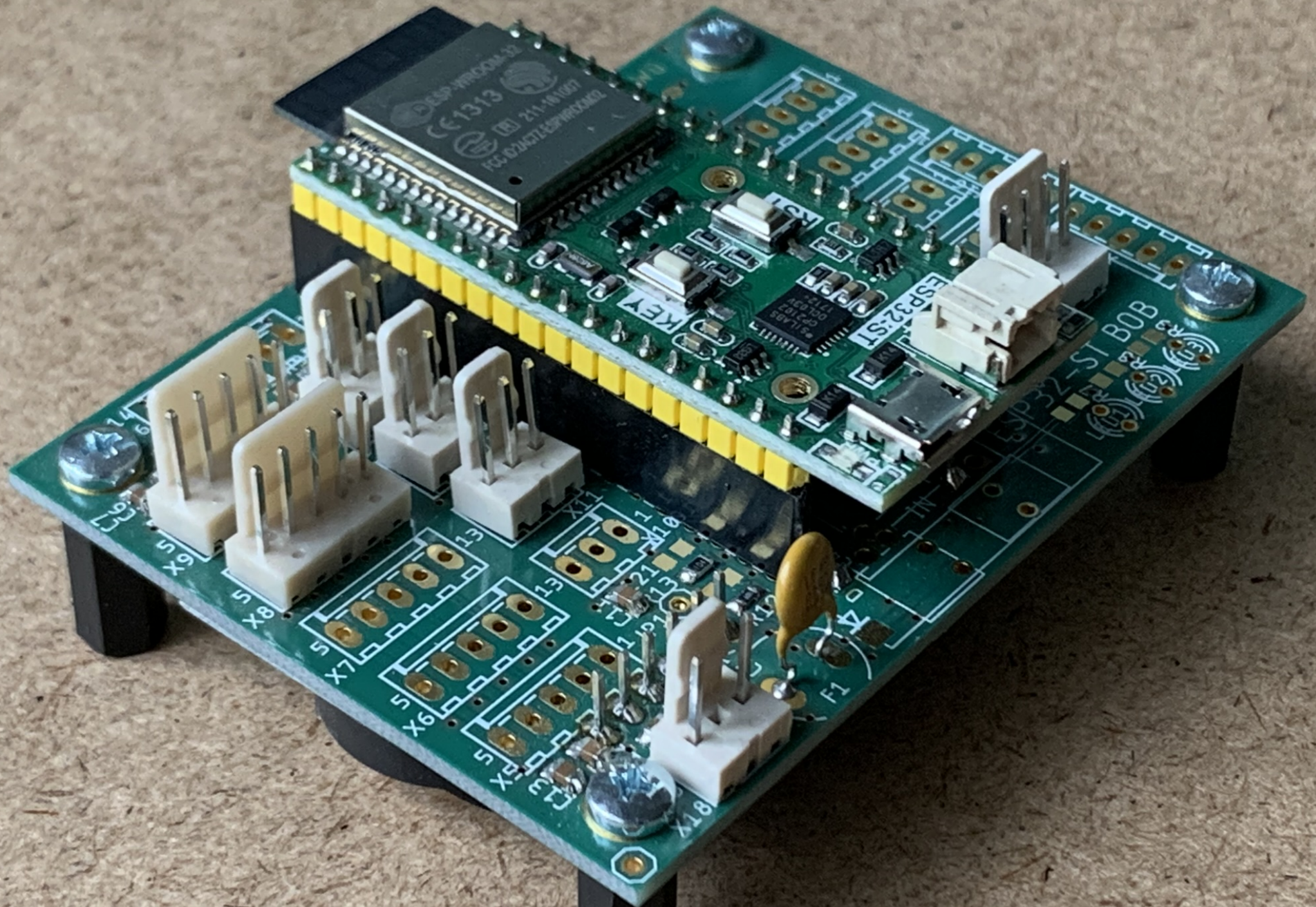










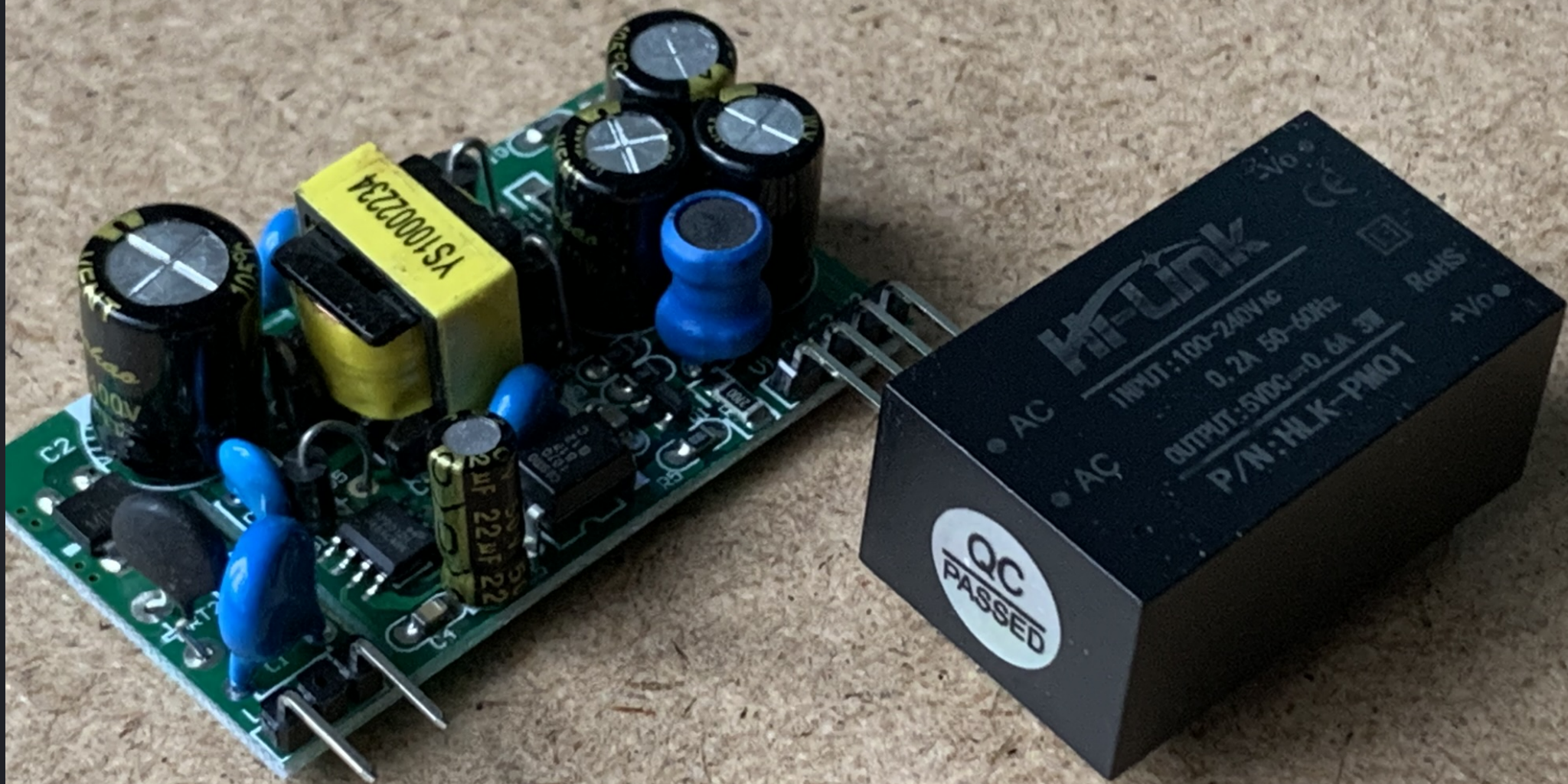


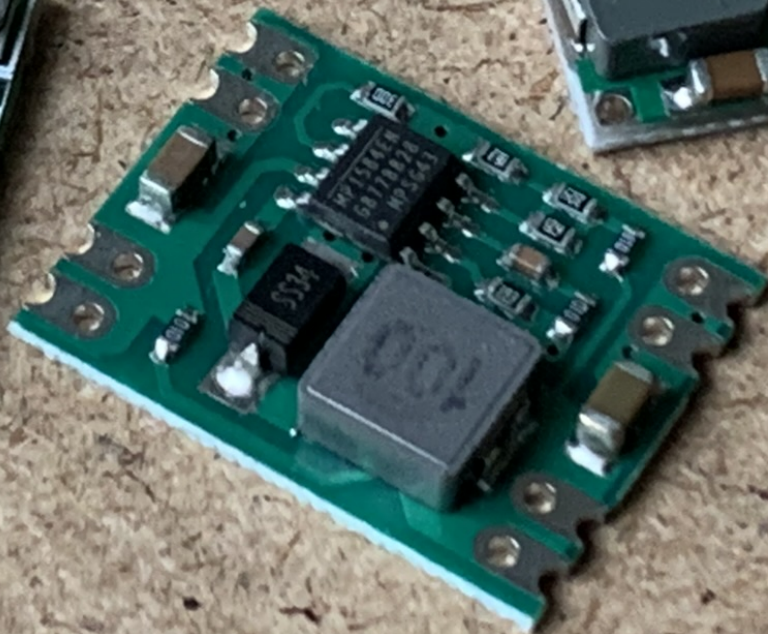
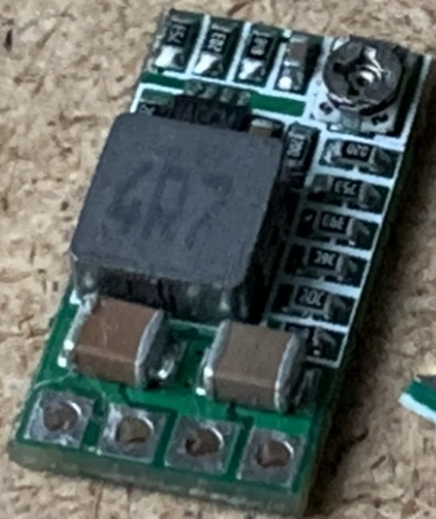
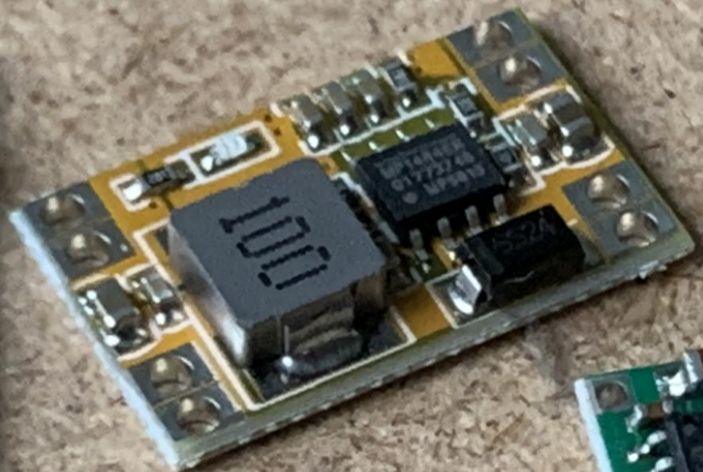
Spannungsversorgung?

Lösungen:

- Steckernetzteil
- Netzteil-Modul
- DC/DC-Wandler







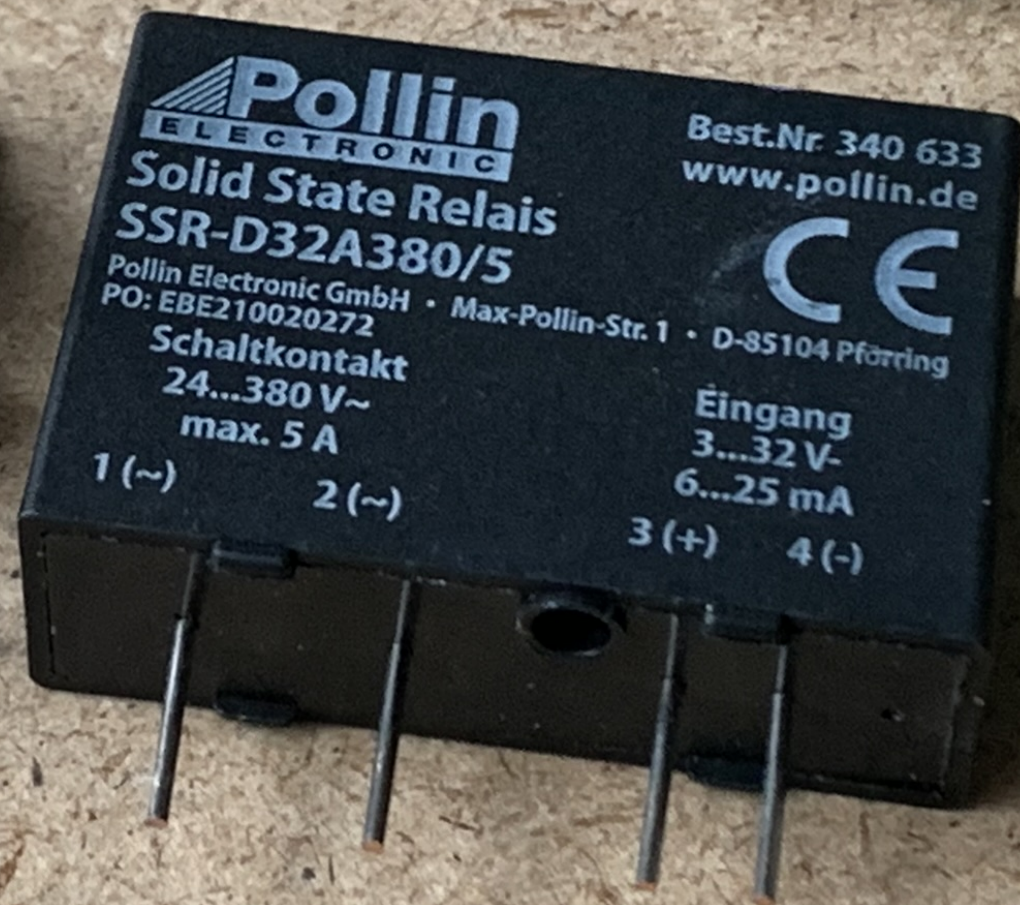
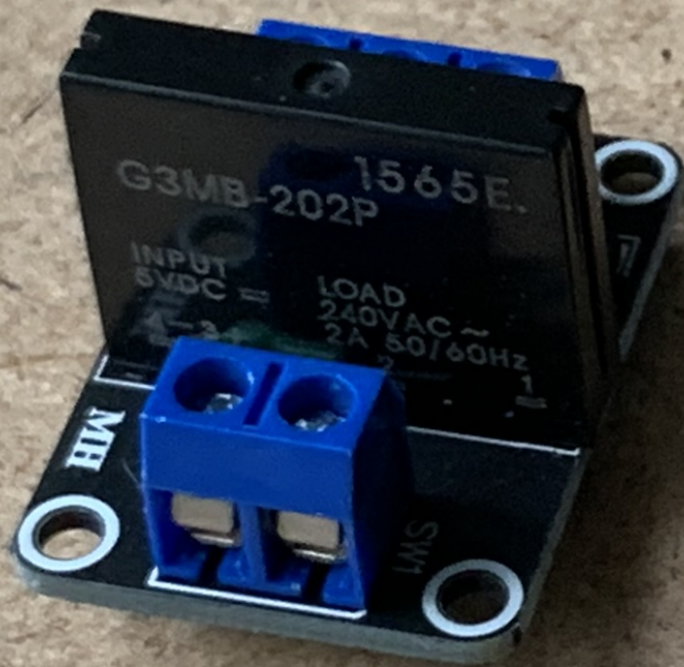
Sensoren / Aktoren?

Sensoren:

- Temperatur -> DS18C20
- Temperatur+Feuchte+Luftdruck -> BME280
- ...

Aktoren:

- 230V schalten -> Solid-State-Relais (SSR)
- ...



Ready-To-Use?

Firmware:

- Tasmota (ESP8266)
- ESP-Easy (ESP8266)
- ezPiC (ESP32) [in Entstehung]
- ...

Daten - wohin?

Schnittstellen:

- REST-API (Client)
- REST-API (Server)
- MQTT
- LoraWAN
- SD-Karten-Logger
- ...

REST-API (Client)

```
import requests

res = requests.get('http://api.openweathermap.org/data/ ...')
if res.status_code == 200:
    jres = res.json()
    print(jres)
```

REST-API (Server)

```
import asyncio

async def handle_connection_http(reader, writer):
    line = await reader.readline()
    parts = line.split(b' ', 2)

    while 1:
        line = await reader.readline()
        if line == b'\r\n': # end of header
            break

        await writer.write(b'HTTP/1.0 200 OK\r\n\r\n')
        await writer.write(HTML_FORM + b'\r\n\r\n')
        await writer.aclose()

aserver_http = asyncio.start_server(handle_connection_http, '0.0.0.0', 80)
loop.create_task(aserver_http)
loop.run_forever()
```

MQTT

```
from umqttsimple import MQTTClient

def sub_cb(topic, msg):
    print((topic, msg))
    if topic == b'notification' and msg == b'received':
        print('ESP received hello message')

def connect_and_subscribe():
    global client_id, mqtt_server, topic_sub
    client = MQTTClient(client_id, mqtt_server)
    client.set_callback(sub_cb)
    client.connect()
    client.subscribe(topic_sub)
```

HTTP-Server

```
import tinyweb

app = tinyweb.server.webserver()

@app.route('/')
async def index(request, response):
    await response.send_file('static/index.html')

@app.route('/test')
async def index(request, response):
    await response.start_html()
    await response.send('<html><body><h1>Hello, world! ...</h1></html>\n')

app.run(host='0.0.0.0', port=80)
```

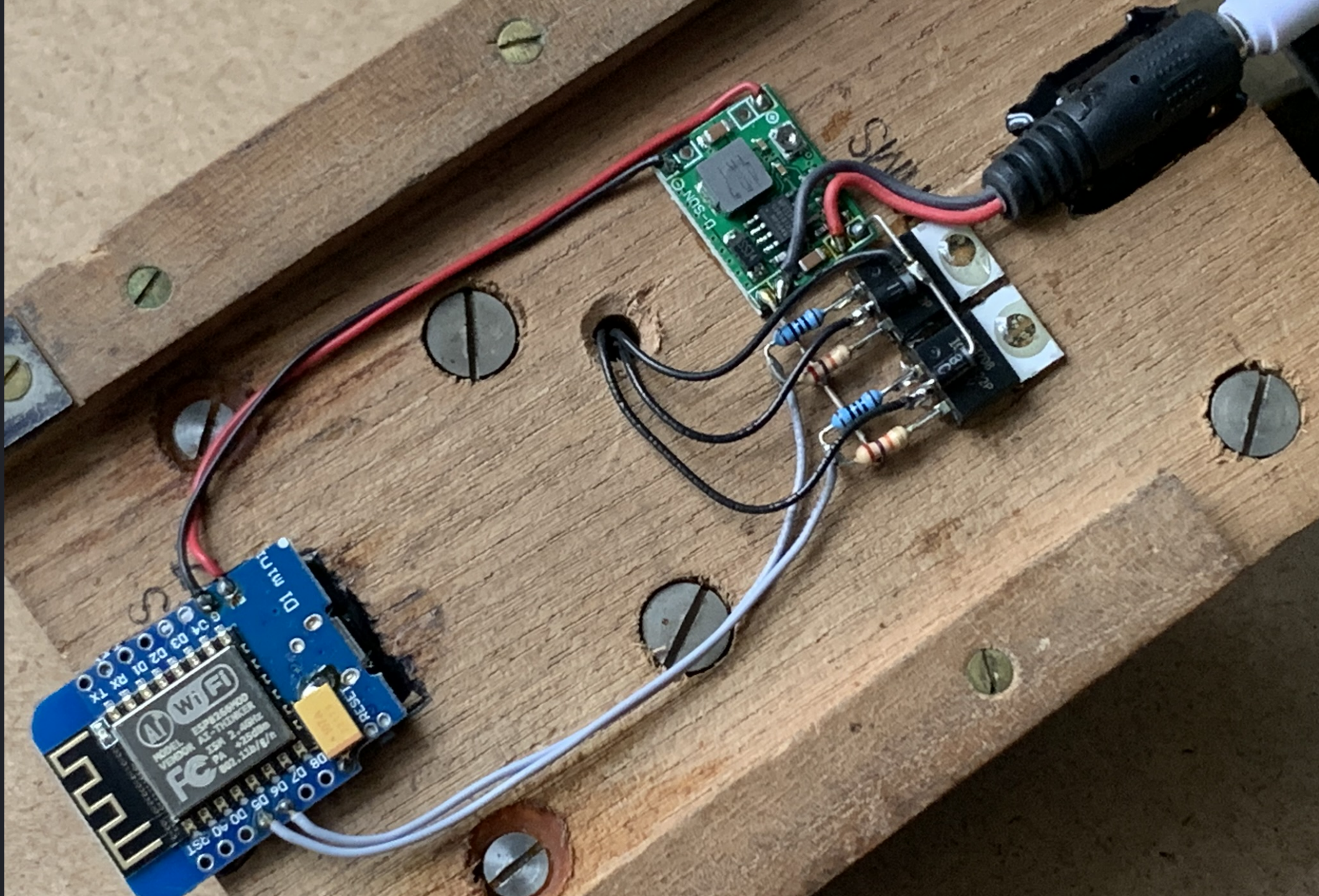
Beispiele?

IoT-Spielereien:

- Wetter-Display
- Stundenschlag
- Bon-Drucker









Fragen?

jk@nerd2nerd.org