

# Projektaufgabe Embedded Systems SS2024

Tim Krüger, 1197031

13.08.2024

## Inhaltsverzeichnis

|  |          |
|--|----------|
| <b>1 Allgemeines</b>                     | <b>2</b> |
| 1.1 Persönliche Angaben . . . . .        | 2        |
| 1.2 Eigenständigkeitserklärung . . . . . | 2        |
| <b>2 Einleitung</b>                      | <b>3</b> |
| 2.1 Motivation . . . . .                 | 3        |
| 2.2 Aufgabenstellung . . . . .           | 3        |
| <b>3 Installationsanleitung</b>          | <b>4</b> |
| 3.1 Projektstruktur . . . . .            | 4        |
| 3.2 Netzwerkboot . . . . .               | 4        |
| 3.3 Buildroot . . . . .                  | 5        |
| 3.4 Kernel . . . . .                     | 5        |
| 3.5 WLAN-AP . . . . .                    | 5        |
| 3.6 Hardwareaufbau . . . . .             | 5        |
| 3.7 Gerätetreiber . . . . .              | 5        |
| 3.8 MQTT-Konfiguration . . . . .         | 5        |
| 3.9 Anderes . . . . .                    | 5        |
| <b>4 Systemtest</b>                      | <b>5</b> |
| 4.1 Testplan . . . . .                   | 5        |
| 4.2 Komponententest . . . . .            | 5        |
| 4.2.1 Boot und Entwicklung . . . . .     | 5        |
| 4.2.2 LED-Treiber . . . . .              | 5        |
| 4.2.3 WLAN . . . . .                     | 5        |
| 4.2.4 MQTT . . . . .                     | 7        |
| 4.3 Test des Gesamtsystems . . . . .     | 8        |
| <b>5 Zusammenfassung</b>                 | <b>8</b> |

# 1 Allgemeines

## 1.1 Persönliche Angaben

Name: Tim Krüger

Matrikelnummer: 1197031

Studiengang: Master Informatik

Datum: 13.08.2024

## 1.2 Eigenständigkeitserklärung

Eidesstattliche Erklärung  
zur Projektarbeit Eingebettete Systeme im SS2024

Name:

Matrikelnummer:

Ich versichere durch meine Unterschrift, dass die vorgelegte Arbeit ausschließlich von mir erstellt und verfasst wurde. Es wurden keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt.

Krefeld, 13.08.24

Ort, Datum

Krüger

Unterschrift

Abbildung 1: Eigenständigkeitserklärung

## **2 Einleitung**

### **2.1 Motivation**

### **2.2 Aufgabenstellung**

### 3 Installationsanleitung

#### 3.1 Projektstruktur

Als Teil dieses Projekts werden alle notwendigen Dateien (Dokumentation, Configs, Skripte etc.) mitgeliefert. Die Struktur des Dateibaums ist im folgenden dargestellt:

```
Dev/ESY/
|-- buildroot/
|-- configs/
|-- devicetree/
|-- docs/
|   |-- res/
|-- kernel/
|-- modules/
|-- scripts/
|-- target/
`-- txts/
```

- *Dev/ESY/*: Das Root-Verzeichnis des Projekts
  - Bei einer anderen Verzeichnisstruktur müssen die Skripte *post-build.sh* und *post-image.sh* entsprechend angepasst werden
- *buildroot/*: Das Buildroot-Verzeichnis.
  - Genaue Hinweise folgen in Kapitel 3.3
- *configs/*: Verzeichnis für die Buildroot- und Kernel-Konfigurationsdateien
- *devicetree/*: Verzeichnis für den Device-Tree-Blob zur Ansteuerung der GPIOs
- *docs/*: Dokumentation des Projekts inkl. dem Abbildungsverzeichnis *res/*
- *kernel/*: Verschiedenen Kernelversionen, welche im Laufe des Projekts entstanden sind
  - Genaue Hinweise folgen in Kapitel 3.4
- *modules/*: Der Treiber bzw. das Kernelmodul zur Ansteuerung der LED
  - Genaue Hinweise folgen in Kapitel 3.7
- *scripts/*: Verschiedene Skripte (Teil des Entwicklungsprozesses)
- *target/*: Verschiedenen Skripte und Konfigurationsdateien, welche auf den Raspberry Pi kopiert werden
- *txt/*: Verschiedene Textdateien, hauptsächlich zur initialen TFTP-Konfiguration

#### 3.2 Netzwerkboot

Für einen schnellen Entwicklungsprozess (Iterationsgeschwindigkeit + Deployment) wird Netzwerkboot via TFTP verwendet. Der Raspberry Pi zieht sich hierbei, bei korrekter Konfiguration, alle notwendigen Dateien des Bootprozesses von einem TFTP-Server, welcher auf dem Host-Rechner läuft.

Ein solches Setup kann mit den folgenden Schritten ans Laufen gebracht werden:

- Installation und Starten eines TFTP-Servers auf der Host-Maschine
  - Monitoren des Outputs bspw. via: *tail -f /var/log/syslog | grep tftp*
  - Empfehlung: Extensives Logging aktivieren
- Originale Pi4-Bootfiles herunterladen und ins TFTP-Verzeichnis kopieren
- Bootloader des Raspberry Pi's **muss** für Netzwerkboot angepasst werden
  - Dafür muss der Pi entsprechend geflashed werden
- Anpassung der Netzwerkschnittstellen
  - Zuweisung einer statischen IP ans Ethernet-Interface des Host-PC
  - Zuweisung und Konfiguration des Ethernet-Interface auf dem Pi (erfolgt über die Datei */target/interfaces*)

Es ist außerdem empfehlenswert das serielle Interface des Pi's für Output und Debugging zu verwenden. Die notwendigen Konfigurationsdateien werden bereitgestellt:

- Kopieren der Konfigurationsdateien
  - */txt/cmdline.txt* nach */srv/tftp/*
  - */txt/config.txt* nach */srv/tftp/*
- Installation eines beliebigen Terminalemulators, z.B. *Minicom*
  - Starten bspw. via: *sudo minicom -D /dev/ttyUSB0*

- 3.3 Buildroot
- 3.4 Kernel
- 3.5 WLAN-AP
- 3.6 Hardwareaufbau

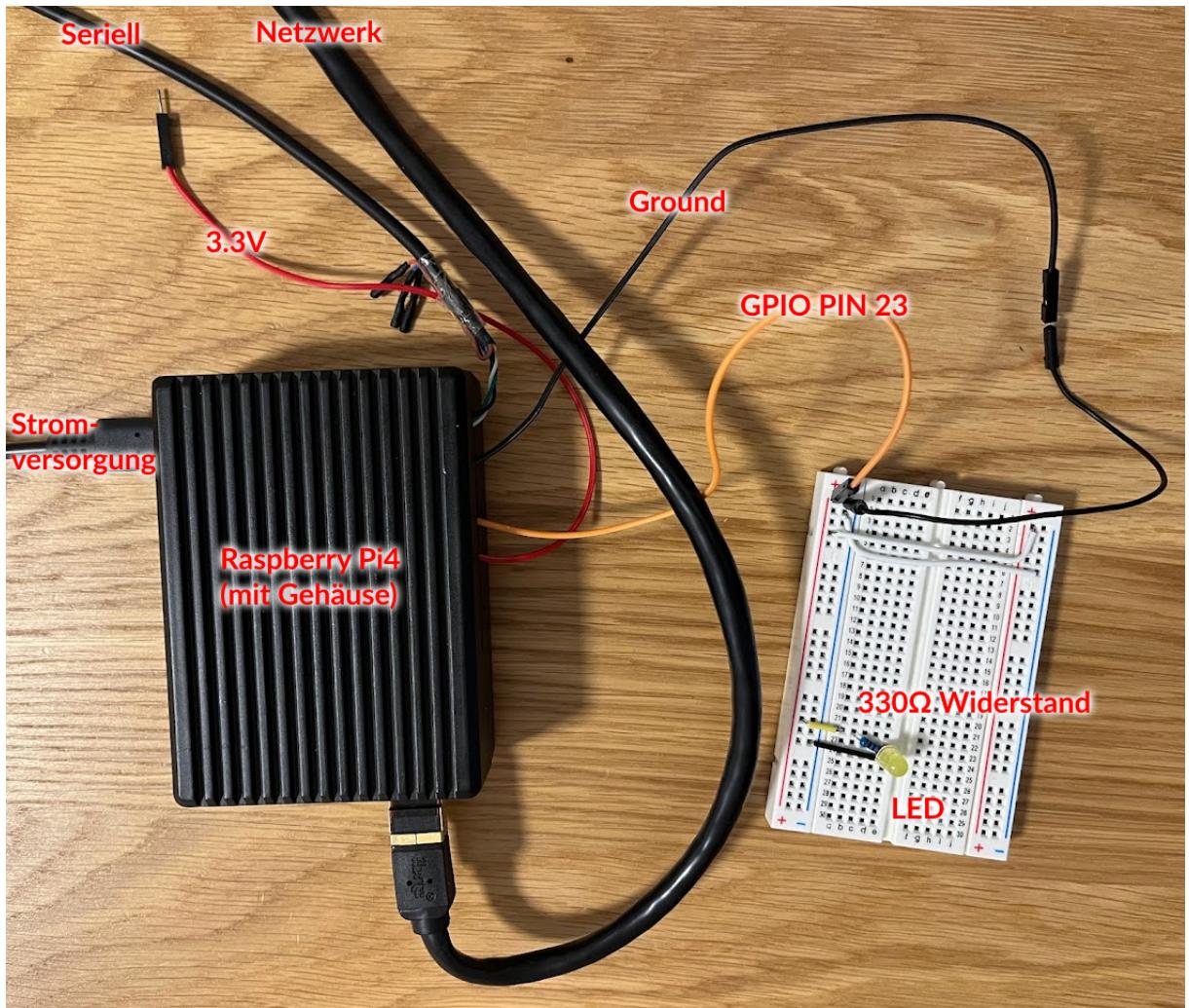


Abbildung 2: Hardwareaufbau

- 3.7 Gerätetreiber
  - 3.8 MQTT-Konfiguration
  - 3.9 Anderes
- ## 4 Systemtest
- 4.1 Testplan
  - 4.2 Komponententest
    - 4.2.1 Boot und Entwicklung
    - 4.2.2 LED-Treiber
    - 4.2.3 WLAN

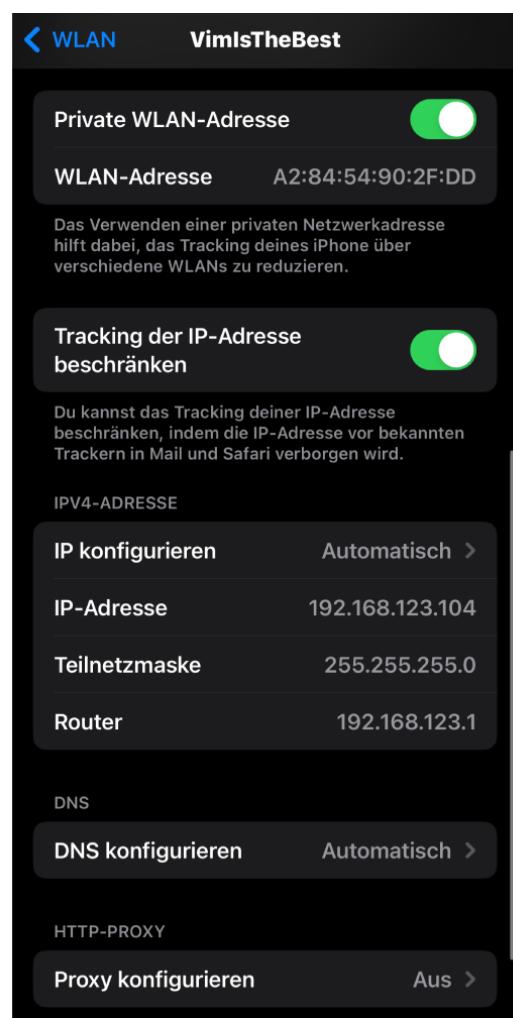


Abbildung 3: WLAN-Test

#### 4.2.4 MQTT

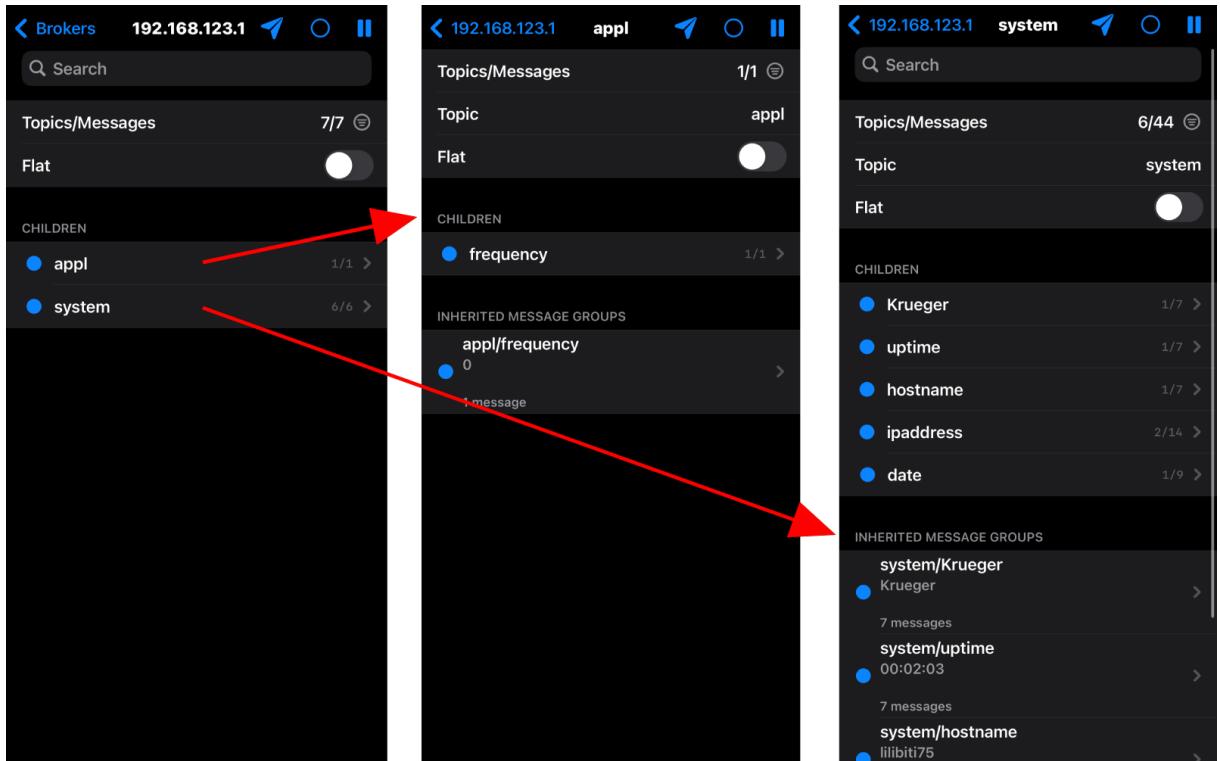
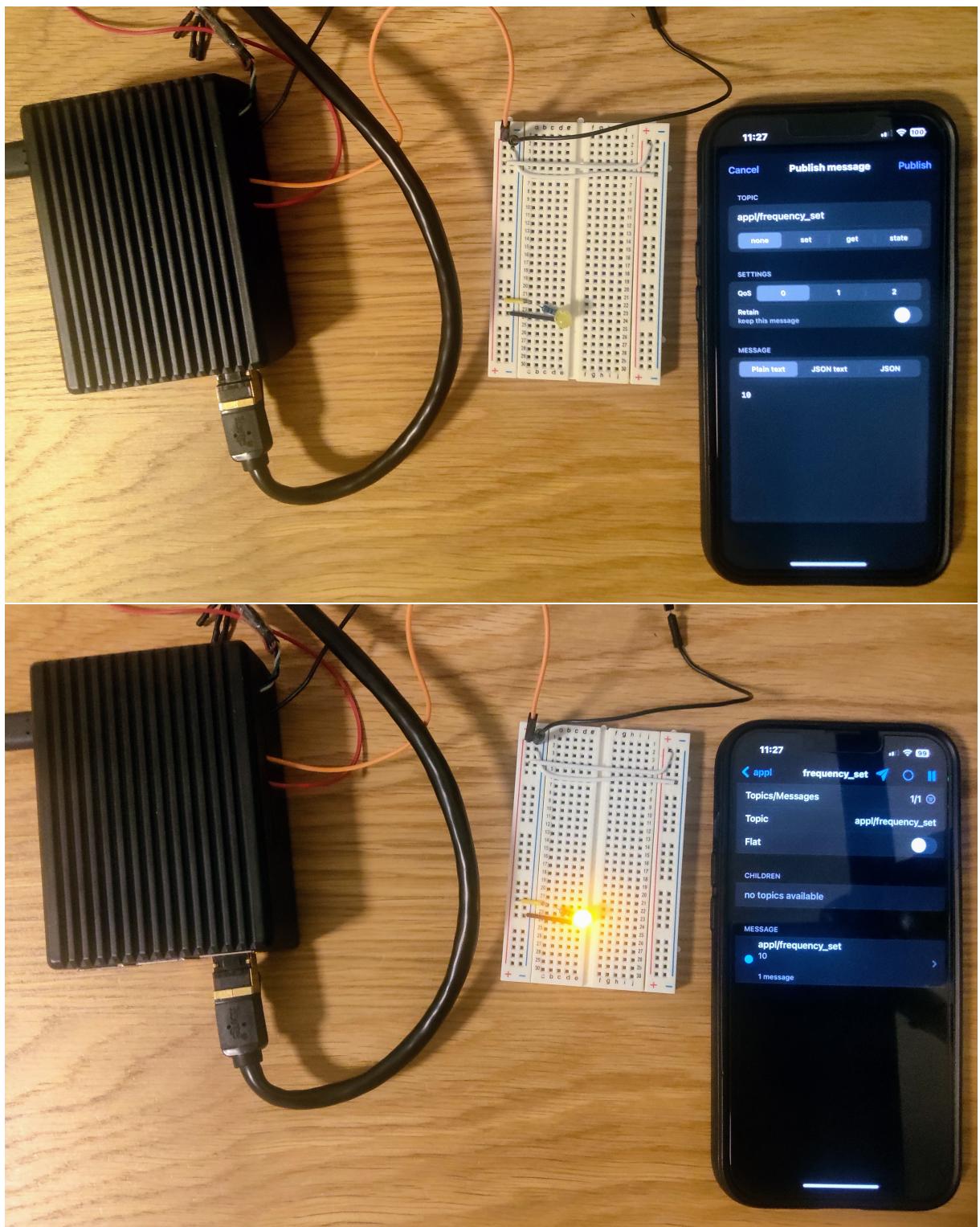


Abbildung 4: MQTT-Test

#### 4.3 Test des Gesamtsystems



#### 5 Zusammenfassung