

# ZQ Programmierung für das IoT - AFBB Dresden

12.-16.06.2023 Alexander Lorz

## Lernziele

Nach Abschluss diese einwöchigen Praktikums sind Sie in der Lage:

- Im Team gemeinsam eine bisher unbekannte komplexe IT-Aufgabe zu bearbeiten, Lösungskonzepte zu entwickeln, diese zu dokumentieren und prototypisch umzusetzen.
- Die dazu notwendigen Teilaufgaben eigenständig zu identifizieren und über mehrere Entwicklungsgruppen und -phasen zu verteilen und zu koordinieren.
- Grundlegende Konzepte des Internet of Things zu benennen und deren Bedeutung für die Entwicklung eines komplexen IT-Systems darzustellen.
- Wichtige Technologien und Protokolle für ein einfaches IoT-System zu benutzen und dafür Schnittstellen zwischen Systemkomponenten zu entwerfen und umzusetzen.
- Einfache Sensoren/Aktoren sowie die passende Backend-Infrastruktur zu deren Entwicklung und Betrieb zu integrieren und ggfs. selbst zu bauen.

## Grundregel: Doku, Doku, Doku - Document All The Things!

Dokumentation und Code via GitHub (ein gemeinsames Repo für Infos + privates Repo für jedes Team).

- <https://github.com/ZQ-AFBB-2023>

**Lösungswege und Entwurfsentscheidungen** (nicht nur die Ergebnisse) dokumentieren!

- “Kochrezepte” schreiben: Tue X, installiere Y, führe folgendes Kommando aus.  
ZIEL: Mit der Anleitung soll die Lösung reproduzierbar sein.  
(Was ist wenn die SD-Karte oder ein Raspberry kaputt geht oder alles gelöscht wird?)
- Hintergrundinformationen (Videos, Blogartikel, Anleitungen) nicht nur verlinken, sondern auch (kurz) zusammenfassen und bewerten: Welcher Inhalt steckt hinter dem Link?  
Funktioniert das gut/schlecht oder ist es vielleicht sogar veraltet?

Aufwand für Doku angemessen halten: Anfangs eine README.md später separate Teile, die aus der README.md heraus verlinkt werden.

- “Digitales Schmierpapier” anlegen: Persönliche Datei notizen.txt mit unstrukturierten Informationen, Links, Kommandos, Notizen. Diese Inhalte nach und nach strukturieren und in die Dokumentation einfügen. Veraltete Informationen korrigieren oder löschen.
- Bearbeitbare Formate für Grafiken ablegen, nicht nur PNGs/JPEGs!

- **Keine persönlichen Informationen und keine Zugangsdaten auf Github!**  
→ Jedes Team pflegt einen Password-Store (Papier reicht für das Praktikum).
- Urheberrechte beachten! Fremde Inhalte (Texte, Bilder) nur dann verwenden, wenn zulässig. Diese korrekt kennzeichnen und zitieren (Quelle, Lizenz). Quellen müssen anhand der Angaben auffindbar sein.

## Arbeitsprotokoll: “Am Ende des Tages”

- Wer wollte was machen? Wer hat wie viel Arbeit in was investiert? Was ist am Ende tatsächlich dabei herausgekommen? Ziel: Reflexion über das Erreichte
- Arbeitsprotokoll: Zeit von-bis; Beteiligte; Personenstunden; Aufgabe; Ergebnis(se)
- Pro Team, tageweise, ein(e) Verantwortliche(r) je Team
- Enthält personenbezogene Daten! Nicht ins Git! → Papier reicht für das Praktikum

## ZIEL1: Dinge ausprobieren + WLAN einrichten

- WLAN SSID **ZQ\_IOT\_120** Passwort siehe Aushang
- Zugriff auf die mitgebrachten Geräte per Weboberfläche ausprobieren  
→ LED-Streifen: <http://192.168.120.81> (Achtung, zeitgleicher Zugriff begrenzt)
- Zugriff auf GitHub sicherstellen
- Installieren des Pakets `mosquitto-clients`, Steuerung auf der Kommandozeile mit `mosquitto_pub` und `mosquitto_sub` ausprobieren. Weiter Infos, siehe Github-Repo.
- Das kann recht schnell unübersichtlich werden und es empfiehlt sich zusätzlich die Benutzung einer grafischen Benutzeroberfläche, z. B. <http://mqtt-explorer.com/>.
- **EIGENES WLAN einrichten**
- **Dokumentiert** Vereinbarungen zur Vergabe von IP-Adressen: Welche Gruppe benutzt welche IP-Ranges? In welchem Bereich werden Adressen automatisch vergeben, welche sind reserviert für Geräte? Wer hat die Übersicht und trifft Entscheidungen (eine Person, nicht mehrere)? Wo findet sich eine aktuelle Übersicht?  
→ Einfache Lösungen bevorzugen, **Verantwortliche** benennen.  
→ Statische IPs, DHCP-Range, Static DHCP leases (ggfs. per Gruppe)  
→ Gut merkbare Lösungen: z. B. Gruppe 1 bekommt x.y.z.111-119, Gruppe 2 x.y.z.121-129  
→ Kein Over-Engineering! Dinge einfach halten. Brauchen wir wirklich Subnetze?

## ZIEL2: Raspberry Pi einrichten

### Installieren von Raspberry Pi OS (32-bit)

- Diesen Abschnitt zuerst vollständig durchlesen.

- Installation siehe <https://www.raspberrypi.com/software/operating-systems/>  
Raspberry Pi OS with desktop - Release date: May 3rd 2023 - System: 32-bit
- Inzwischen geht das recht einfach mit dem Raspberry Pi Imager  
<https://www.raspberrypi.com/software/>
- **Vorher im Konfigurationsmenüs des Imagers einen Hostnamen vergeben, SSH-Zugang ermöglichen und initiale Passwörter vergeben. Wichtig: Als Nutzernamen den Standardwert pi verwenden!**
- Nach dem ersten Start (dauert etwas) Initiale Konfiguration
  - **statische IP vergeben**  
(<https://www.raspberrypi.org/documentation/configuration/tcpip/>)  
**Vorzugsweise per statischem DNS-Lease auf dem Router vergeben!**
  - Locale, Zeitzone, SSH-Zugang, ... ggfs. anpassen
  - **initiales Update/Aktualisierung** (sudo apt update, sudo apt upgrade)

## ZIEL3: IOTStack installieren und einrichten

Lesen Sie diesen Abschnitt und das Tutorial auf [https://sensorsiot.github.io/IOTstack/Basic\\_setup/](https://sensorsiot.github.io/IOTstack/Basic_setup/) **vollständig**. Beginnen Sie erst danach mit der Installation.

- Install-Script laufen lassen, dieses installiert den IoT-Stack und Docker. Dabei **patch 1 und patch 3** (siehe Anleitung) nicht vergessen. Patch 2 **nicht** anwenden. Das dauert eine Weile, anschließend den Pi neu starten.

Benötigte Bestandteile (container) installieren

- [X] espruinohub
- [X] grafana
- [X] influxdb (nicht influxdb2!)
- [X] mosquito
- [X] nodered (issue: → options, build add-on-file)
- [X] portainer-ce

Stack erstmalig mit `docker-compose up -d` starten. Das wird eine Weile dauern, da zuerst die Container-Images heruntergeladen werden. Die einzelnen Komponenten sollten nun erreichbar sein:

- Grafana: Port 3000 admin/admin (Passwort beim erstem Login neu vergeben)
- Node-RED: Port 1880 (ohne Passwort)
- Portainer CE: Port 9000 (Passwort bei erstem Login vergeben)
- MQTT-Broker auf Port 1883

## ZIEL4a: Erste Schritte mit MQTT und Node-Red

Mit dem MQTT-Explorer (<http://mqtt-explorer.com/>) sollten Sie jetzt bereits auf „Ihren“ MQTT-Broker zugreifen können. Unter der Topic ble (für Bluetooth Low Energy) sehen Sie mit Sicherheit bereits zahlreiche Geräte. Lassen Sie sich vom Betreuer einen BLE-Beacon geben und finden Sie heraus, welche Daten der Beacon auf sendet.

Jetzt können Sie damit beginnen, die Anwesenheit des Beacons in einem Node-Red Dashboard zu visualisieren:

- Erstelle Sie in Node-Red einen neuen Flow.
- Ziehen Sie einen mqtt-Inputknoten und einen Debug-Outputknoten auf die Seite, und verbinden Sie sie miteinander.
- Doppelklicken Sie auf den mqtt-Knoten, und vergewissern Sie sich, dass als Server mosquitto:1883 verwendet wird. Sie müssen einen neuen Server hinzufügen, wenn Sie Node-RED zum ersten Mal verwenden.
- Stellen Sie sicher, dass Topic auf /ble/# steht - # ist ein Platzhalter in MQTT
- Klicken Sie nun oben rechts auf Deploy und anschließend im rechten oberen Bereich auf das Debug-Panel (Bug/Käfer-Symbol).
- Die Nachrichten aller BLE-Geräte im Umfeld zu sehen ist nicht besonders hilfreich. Finden Sie heraus, wie sie dies auf die Advertise-Nachrichten „Ihres“ Beacon einschränken können. Tip: <http://www.steves-internet-guide.com/understanding-mqtt-topics/>
- Wenn Sie das geschafft haben fügen Sie einen neuen function-Knoten mit der Funktion

```
msg.payload = msg.payload.rssi;
return msg;
```

sowie einen chart-Knoten hinzu, konfigurieren diese und binden den Chart-Knoten in ein Dashboard ein.
- Falls Ihnen das nicht intuitiv gelingt, nutzen Sie bitte folgendes Video um sich mit Node-Red vertraut zu machen: <https://media.ccc.de/v/froscon2018-2209-node-red>  
Sie können die ersten 15 Minuten (Installation von Node-Red) überspringen.
- **Sind Sie die erste Grupp, die dieses Ziel erreicht, so sprechen Sie bitte den Betreuer an.**
- **HINWEIS:** espruinohub publiziert standardmäßig auf "/ble" statt auf "ble", siehe <https://github.com/espruino/EspruinoHub/issues/38>. Das ist eigentlich nicht gut, ist aber einmal so. Siehe "Never use a leading forward slash" <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/>.

## ZIEL4b: Enocean-Devices einrichten

Die Geräte wurden uns vom IBM Watson Center Munich und vom IBM Innovation Studio leihweise zur Verfügung gestellt. Wir haben 2 funktionsfähige Gateways, ggfs sollten alle Gruppen ein Gateway gemeinsam nutzen.

# ZIEL5: Vorträge zu IoT-Grundlagen und Technologien

Von jeder Gruppe: 10-15 minütiger Vortrag (PPT) zu je einem Thema. Bitte sprechen Sie mit dem Betreuer ab, welches Thema Ihre Gruppe übernimmt (5a oder 5b).

## **ZIEL5a: Thema 1 “Potenziale und Risiken des Internet of Things” als Vorbereitung auf die Abschlusspräsentation**

**Zeitplanung: Gliederung bis Tag 2 13:00; Vortrag Ende Tag 2 fertig; Präsentation zu Beginn von Tag3**

**Einstiegs-Quellen Teil 1:** Was ist “dieses IoT” überhaupt? Warum ist es interessant?

Internet of Things(IoT) Applications | IoT Tutorial for Beginners

ENGLISCH ca. 17min

<https://www.youtube.com/watch?v=OfGxbxUCa2k>

Internet Of Things (IoT) In 10 Minutes | What Is IoT And How It Works

ENGLISCH ca. 9 min

<https://www.youtube.com/watch?v=Fj02iTrWUx0>

*Finden Sie weitere Internetquellen (Blogs, Vorträge, Videos, ...), die eine grundlegende Einführung in das Thema bieten und erstellen Sie eine kurze Zusammenfassung der besten (maximal 3) Quellen. Welche Inhalte werden vermittelt? Warum ist gerade diese Quelle besonders gut?*

**Einstiegs-Quellen Teil 2:** Risiken von IoT

How dangerous are IOT devices? | Yuval Elovici | TEDxBGU

ENGLISCH 13 min

Privacy: [https://www.youtube.com/watch?v=vgoX\\_m6Mkko](https://www.youtube.com/watch?v=vgoX_m6Mkko)

Internet of Things (IoT) Privacy & Security2018-10-26 Ralf Schlatterbeck PrivacyWeek 2018

DEUTSCH 43 min

<https://media.ccc.de/v/pw19-251-internet-of-things-iot-privacy-security>

Best of IoT Fails - When reality beats science fiction

ENGLISCH ca. 30 min

[https://media.ccc.de/v/SHA2017-163-best\\_of\\_iot\\_fails](https://media.ccc.de/v/SHA2017-163-best_of_iot_fails)

BoingBoing - The internet of shit

<https://boingboing.net/tag/internet-of-shit>

*Teilen Sie die Arbeit auf. Nicht jede(r) muss jede Quelle studieren. Diskutieren Sie Ihre Erkenntnisse im Anschluss in der Gruppe und entwerfen Sie gemeinsam den Vortrag. Tipp: ggfs. auf YouTube die Untertitel aktivieren.*

## **ZIEL5b: Thema 2 “MQTT und Low-Code-Programmierung mit Node Red“**

**Zeitplanung: Präsentation mit Live-Demo im Laufe von Tag 2, Ziel ist die Unterstützung der anderen Gruppen**

Machen Sie sich mit NodeRed und MQTT vertraut, finden Sie passende Internetquellen und erstellen Sie teamübergreifend einen kurzen Vortrag mit Live-Demo (ca. 10-15 Minuten) mit dem Sie eine Einführung in MQTT und Node-Red geben können. Gute Quellen zum Einstieg sind:

- MQTT - Die Sprache im Internet der Dinge DEUTSCH  
<https://media.ccc.de/v/froscon2018-2206-mqtt>
- Node-Red- Der Fluss der Dinge DEUTSCH  
<https://media.ccc.de/v/froscon2018-2209-node-red>

Steuern Sie vorhandene Geräte mittels eines Node-RED Dashboards. Dieses sollte auch den aktuellen Zustand der Geräte anzeigen. Gehen Sie insbesondere auf MQTT-Topics und Wildcards ein und erklären Sie den anderen wie sie mit Node-Red Anwendungen entwickeln können.

## **Ziel6: Eigenes IoT-Device bauen | Anwendungsszenarien**

**Stimme Sie sich im Team ab, welche Aufgabe Sie lösen wollen. Falls sie genügend Zeit zur Verfügung haben können Sie mehrere Geräte bauen bzw. die Aufgabenstellung in Abstimmung mit den Betreuern verändern.** Die Aufgabe 6a kennen Sie vielleicht schon aus den ZQs der letzten Jahre. Sie erfordert (ein klein wenig) mehr Aufwand beim Dimensionieren und Umsetzen einer Schaltung zum Anschluss einer einfachen einfarbigen Lichterkette. 6b ist bezüglich der Hardware sehr einfach, erfordert aber die selbständige Einarbeitung. Bei 6c können Sie ihrer Kreativität freien Lauf lassen (soweit wir die dafür notwendige Hardware da haben).

### **ZIEL6a: Einfarbiges Tasmota-SmartLight mit ESP8266**

Installation von esptool zum flashen der Firmware, siehe <https://tasmota.github.io/docs/Esptool/>

- Achtung: Das Binary bzw. Linux-Package esptool funktioniert nicht richtig, daher die Python-Variante: esptool.py installieren:
  - `sudo apt-get install python3-pip`
  - `sudo pip3 install esptool`
- Zum Programmieren GPIO0 (=D3) des ESP8266 auf GND legen!
  - Pinout-Referenz: <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>
  - Flash löschen `sudo esptool.py -p /dev/ttyUSB0 erase_flash`
  - Tasmota-Firmware herunterladen <https://github.com/arendst/Tasmota/releases/>
  - tasmota.bin (recommended release binary) für ESP8266 verwenden

- Firmware flashen mit  

```
sudo esptool.py -p /dev/ttyUSB0 write_flash --flash_size 1MB
0x0 ./tasmota.bin
```
- Initiale Konfiguration:
  - Wifi entsprechend <https://tasmota.github.io/docs/Getting-Started/#initial-configuration>
  - IP-Reservierung via DHCP einrichten
- GPIOs in der Firmware konfigurieren
  - Configure Template: neues Template, basierend auf Generic(18) für das Modul anlegen
  - Configure Module: mit dem neuen Template D2 GPIO4 als PWM 1 definieren
  - Testen: Messgerät oder Oszi an GPIO 4 (D2) anschließen.
- (kleine) LED anschließen:
  - Dimensionieren Sie einen passenden Vorwiderstand!
  - Was müssen Sie dazu alles wissen?
- Lichterkette anschließen:
  - Sie kennen weder Soll-Spannung noch Strombedarf der Lichterkette. Wie gehen Sie vor?
  - Schlagen Sie eine passende Schaltung zum Anschluss an den ESP8266 vor und setzen Sie diese mit Unterstützung der Betreuer um.

## **ZIEL6b: WLED-SmartLight mit mehrfarbigen LEDs**

Machen Sie sich selbständig mit dem Projekt WLED <https://kno.wled.ge/basics/getting-started/> vertraut, und steuern Sie mit einem ESP32 oder ESP8266 einen mehrfarbigen LED-Streifen an.

Beachten Sie die Strombegrenzung des Netzteils, und stimmen Sie sich vor der Inbetriebnahme mit den Betreuern ab bzw. bitten Sie diese um Unterstützung wenn Sie nicht weiterkommen.

## **ZIEL6c: Einbindung weiterer Sensoren oder Aktoren**

Stimmen Sie sich mit den Betreuern ab, welche weiteren Sensoren oder Aktoren (z. B. Schalter, Feuchtigkeits- oder Lichtsensoren, Servos, ...) sie per MQTT abfragen bzw. Steuern wollen. Unterbreiten Sie Lösungsvorschläge, wie diese z. B. mit Tasmota angeschlossen werden können.

# ZIEL8: Abschlusspräsentation

Stellen Sie Ihre Ergebnisse und Lernerfolge in einer Abschlusspräsentation vor.

**Zeitplanung: Gliederung bis Tag 4 13:00; fertiger Vortrag Ende Tag 4; Präsentation an Tag 5 zum Abschluss**

## Weitere (mögliche, optionale) Aufgaben

- Reichweitentest (Indoor) der EnOcean-Geräte
- Wetterdaten mit NodeRed beziehen und auf dem Dashboard anzeigen (leicht)
- Wetterdaten sowie Status der IoT-Geräte mit NodeRed aufbereiten, in eine InfluxDB schreiben und mit Grafana visualisieren (mittel-anspruchsvoll)
  - Getting started: <https://www.youtube.com/watch?v=JdV4x925au0>
  - Grafana und InfluxDB sind im IOTStack bereits enthalten und brauchen nicht zusätzlich installiert zu werden
  - Präsentieren Sie in einem kurzen Vortrag den anderen Teams wie sie vorgegangen sind.