

Hochschule RheinMain
Fachbereich ITE
Studiengang EE-CS

Scientific Project

Entwicklung eines Zigbee Praktikums unter Verwendung von OpenSource Technologie

verfasst von **Benedikt HEUSER**
Matrikelnummer 105320

am 25.04.2022

Inhaltsverzeichnis

Kapitel 1

Einführung

In diesem Projekt soll ein Praktikumsversuch für die Vorlesung Internet of Things für Professor Jürgen Winter entwickelt werden. In dem Versuch soll das Verhalten des ZigBee Protokolles untersucht werden. Es wird ein kleines ZigBee Netz mit einigen wenigen Komponenten aufgebaut und die Kommunikation mitgeschnitten. Anschließend wird die Kommunikation mit dem Protokollanalyse Werkzeug Wireshark untersucht. Es wird ein “LabGuide “ geschrieben, dass Schritt für Schritt durch den Versuch führt.

1.1 Anforderungen an die Praktikumsarbeit

Die Anforderungen an der Versuch werden an dieser Stelle definiert, um in dieser Dokumentation darauf Bezug nehmen zu können.

- **A010** - Der Versuch soll an einem Tag durchführbar sein.
- **A020** - Der Versuch soll kein Vorwissen in Linux voraussetzen
- **A030** - Der Versuch setzt Vorwissen in Paketorientierten Datenübertragung voraus.
- **A040** - Der Versuch setzt Vorwissen in der Bedienung von Wireshark voraus.
- **A050** - Der Versuch soll zu Hause und in der Hochschule durchführbar sein.
- **A060** - Studenten sollen eine Versuchsbeschreibung sowie alle nötigen Utensilien erhalten. Im Heimversuch müssen KVM-Komponenten von den Studenten selbst gestellt werden.
- **A100** - Der Versuch soll automatisch auf den Raspberry ausgerollt werden können.
- **A110** - Es wird, bis auf den Ausrollvorgang, keine Internetverbindung benötigt.
- **A120** - Es soll ausschließlich gewartete und quelloffene Software zum Einsatz kommen.
- **A200** - Der Versuch soll die Grundlagen eines Mesh-Netzwerkes vermitteln.

- **A210** - Es soll die Funktionsweise des Joinings, des Routings, des Bindings sowie der Gruppenbildung untersucht werden.
- **A210** - Es sollen die implementierten Sicherheitsmechanismen untersucht und bewertet werden.

Kapitel 2

Übersicht Technologien

2.1 IoT Funkprotokolle

Aktuell gibt es mehrere Funkprotokolle, welche im Bereich IoT relevant sind. Dazu gehören:

- **Wlan** Wlan ist ein in jedem Haushalt vorhandener Standard, der überwiegend für die Anbindung mobiler Geräte an den Internetrouter dient. Dies macht es naheliegend, auch smarte Geräte per WLAN einzubinden. Wlan ist allerdings nicht auf eine geringe Leistungsfähigkeit der Hosts, in Bezug auf Rechen- und Sendeleistung ausgelegt. Die ist gerade für Batteriebetriebene Geräte ein enormer Nachteil. Zusätzlich ist es oft nicht gewünscht, Smarte Devices an ein Netzwerk mit Internetzugang anzuschließen
- **Bluetooth** Ebenso wie Wlan hatte Bluetooth auch schon vor dem IoT Boom eine erhebliche Verbreitung. Durch Implementierung des Standard Bluetooth LE ist auch die Leistungsfähigkeit von Devices hier eine geringere Problematik. Bluetooth ist aber nicht für hohe Reichweite und viele Geräte konzipiert. Bluetooth hat keine Skalierfähigkeit wenn es darum geht, eine große Menge von Geräten verteilt im Haus zu vernetzen.
- **Z-Wave**
- **ZigBee** Zigbee ist ein auf den 802.40.5 Standard aufbauendes Protokoll, welches grundlegend für die Anbindung vieler Geräte in einem großen räumlichen Areal konzipiert ist. Ein großer konzeptioneller Vorteil ist, dass bei ZigBee ein Mesh Netzwerk aufgebaut wird. Es können auch Geräte angebunden werden, die keine direkte Funkverbindung zum Koordinator haben. Zusätzlich sind Funktionen implementiert, welche das Management einer hohen Anzahl von Devices erleichtert.
- **Thread** Thread ist ein Newcomer. Es basiert ebenfalls auf den 802.15.4 Standard. Ebenso wie ZigBee ist es Meshfähig, ein entscheidendes Unterscheidungsmerkmal ist allerdings, dass die Geräte per IPv6 adressiert werden. Daher sind die Geräte theoretisch ohne die Verwendung einer Bridge aus einem herkömmlichen Ethernet Netzwerk erreichbar und adressierbar.

2.2 Zigbee Anwendungen

2.2.1 Kommerzielle Anwendungen

Amazon Echo

Der Heimassistent Amazon Echo ist der einzige seiner Art, der eine Zigbee Integration hat und damit als Gateway und Koordinator dienen kann. Die Pendanten der Firmen Google, Microsoft und Apple benötigen ein dediziertes Zigbee Gateway. [Ama12]

Phillips Hue

Phillips stellt eine Zigbee Bridge und eine Vielzahl an Devices aus dem Segment Beleuchtung und Steckdosen.

Dresden Electronic

Dresden Electronic bietet Software und Hardware zum Aufbau von Zigbee Netzwerken an. Es gibt Zigbee USB Adapter und RaspberryPi Hats mit ATmega Chips, sowie eine Steuerungssoftware "de-CONZ". Als komplette Produktlinie für den nicht technisch visierten Endkundenmarkt gibt es die Produktparte "Phoscon", hauptsächlich zur smarten Beleuchtung.

Weitere Hersteller

Weitere bekannte Hersteller/Marken mit Zigbee Devices und Gateways:

- **Logitech** - Harmony Hub
- **LIDL** - Silvercrest
- **TUYA** - Smart Life
- **Innr** - ZigBee Bridge
- **SONOFF**
- **homee** - modular Smart Home Central

Nachteil all dieser Lösungen ist, dass die Kompatibilität zu Geräten von Drittherstellern vollständig in der Hand des Herstellers ist. In der Regel ist aus wirtschaftlichen Gründen die Unterstützung konkurrierender Hersteller nicht gewünscht. Es ist sehr mühsam, bei Anschaffung eines dieser Systeme die Kompatibilität anderer Geräte sicherzustellen.

2.2.2 Nicht kommerzielle Anwendungen

Ein großer Vorteil von OpenSource Anwendungen ist, dass diese durch eine Community gepflegt und Geräte von drittherstellern beliebig integriert werden können. Grundlegend ist der Zigbee Standard universell, und die Kompatibilität von Geräten verschiedener Hersteller möglich.

zigbee2Mqtt

zigbee2Mqtt ist ein quelloffenes Projekt auf GitHub, welches aus einer Serveranwendung mit Web-GUI, und einer Firmware für diverse Texas Instruments Chips besteht. Grundlegende Koordinator Fähigkeiten sind auf der Hardware implementiert, Hardware Abstraktionen sowie die Weiterreichung von Nachrichten an ein MQTT Broker sind in der Webanwendung implementiert. Auf der anderen Seite des MQTT Brokers, zur Visualisierung und Steuerung der Devices kann dann Homeassistant oder ioHAB eingesetzt werden. zigbee2Mqtt bietet eine Menge Möglichkeiten, Informationen zu sammeln und direkt Einflussnahme auf die Devices zu nehmen.

ZHA

ZHA ist ein direkt in HomeAssistant integriertes Plugin, um Zigbee Koordinatoren direkt in HomeAssistant einzubinden. Vorteil von ZHA ist, dass die Liste von unterstützter Zigbee-Chips deutlich länger ist. ZHA unterstützt neben Texas Instruments auch Hardware von Dresden Elektronik, Silicon Labs, DIGI und ZiGate. ZHA ist für den Anwender extrem vereinfacht, es sind kaum technische Informationen ersichtlich oder konfigurierbar.

Kapitel 3

Grundlagen

In diesem Kapitel werden alle verwendeten Komponenten und Technologien kurz erläutert.

3.1 ZigBee

Die ZigBee Alliance wurde durch ein Konsortium von Herstellern gegründet, um einen einheitlichen Übertragungsstandard im Bereich Heimautomatisierung voranzubringen. ZigBee basiert zwar auf einem offenem IEEE Standard, bringt aber proprietäre Komponenten mit. ZigBee ist in Form von weiteren Protokollschichten implementiert, welche oben auf IEEE 802.15.4 aufsetzen.

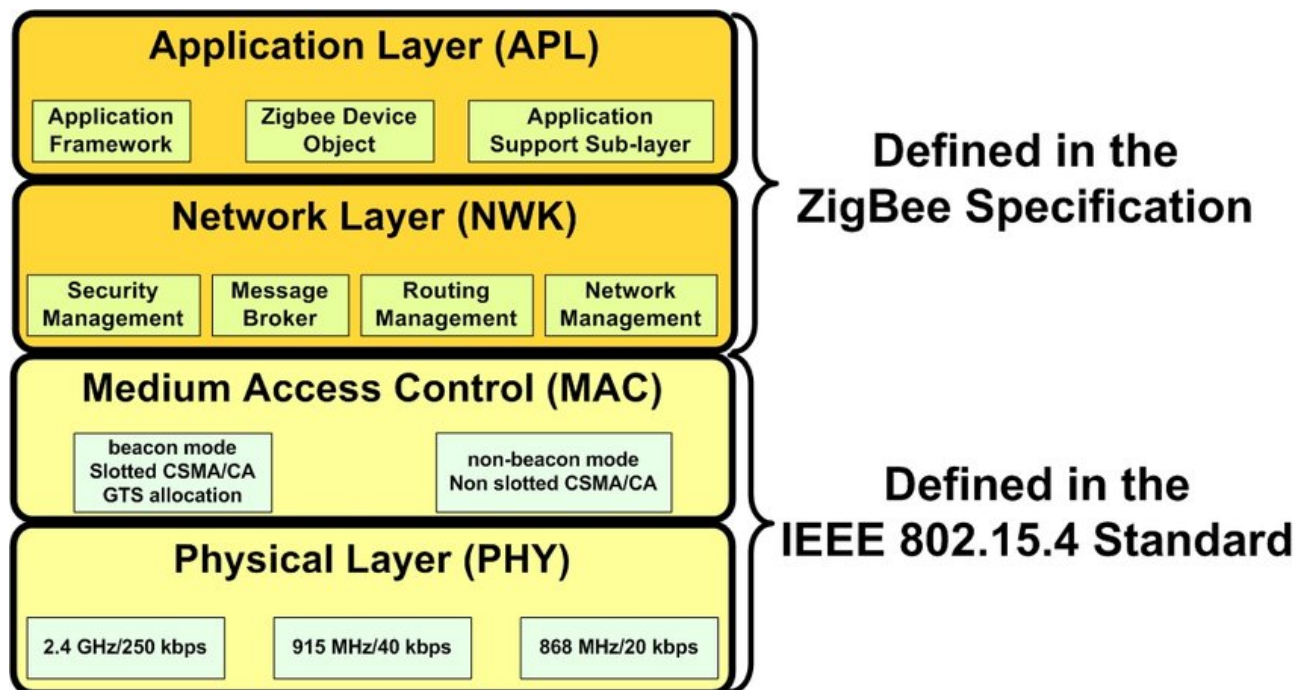


Abbildung 3.1: ZigBee Protocoll Stack

Bildquelle: https://www.researchgate.net/figure/IEEE820154-ZigBee-protocol-stack-architecture_fig2_265150617

Genutzt wird es, um elektrische Geräte im Haushalt, wie zum Beispiel Lampen, Schalter oder Thermostate zur Kommunikation zu befähigen. Markantes Merkmal am Protokoll ist, dass die Geräte keine direkte Funkverbindung zu einem zentralen Controller brauchen. Die einzelnen Geräte können als Router fungieren. Dies sorgt dafür, dass mit einer geringen Sendeleistung ein geografisch großes Gebiet abgedeckt werden kann. Sende- und Empfangsleistung ist vor allem bei kleinen Batteriebetriebenen Geräten der einschränkende Faktor.

3.2 Texas Instruments CC Chips

Texas Instruments bietet ein Spektrum von Microcontrollern, die sich mit entsprechender Firmware als ZigBee Device nutzen lassen können. Als Firmware kann das SZigbee Stack...eingesetzt werden

Die aktuelle Chipfamilie TexasInstruments CC26XX:

Table 3-1. Device Family Overview

DEVICE	PHY SUPPORT	FLASH (KB)	RAM (KB)	GPIO	PACKAGE ⁽¹⁾
CC2650F128xxx	Multi-Protocol ⁽²⁾	128	20	31, 15, 10	RGZ, RHB, RSM
CC2640F128xxx	Bluetooth low energy (Normal)	128	20	31, 15, 10	RGZ, RHB, RSM
CC2630F128xxx	IEEE 802.15.4 Zigbee(6LoWPAN)	128	20	31, 15, 10	RGZ, RHB, RSM
CC2620F128xxx	IEEE 802.15.4 (RF4CE)	128	20	31, 10	RGZ, RSM

(1) Package designator replaces the xxx in device name to form a complete device name, RGZ is 7-mm × 7-mm VQFN48, RHB is 5-mm × 5-mm VQFN32, and RSM is 4-mm × 4-mm VQFN32.

(2) The CC2650 device supports all PHYs and can be reflashed to run all the supported standards.

Abbildung 3.2: Test des Messagebrokers Mosquitto

Als Koordinator werden die leistungsfähigeren Chips aus der 265X Reihe eingesetzt. ZigBee Geräte nutzen in manchen Fällen Bluetooth LE zur Koppelung, daher ist die simultane Unterstützung diesen Protokolls sinnvoll.

6 Device Comparison

Device	RADIO SUPPORT										FLASH (KB)	RAM + Cache (KB)	GPIO	PACKAGE SIZE			
	Sub-1 GHz Prop.	2.4 GHz Prop.	Wireless M-Bus	Wi-SUN®	Sidewalk	Bluetooth® LE	ZigBee	Thread	Multiprotocol	+20 dBm PA				4 x 4 mm VQFN (32)	5 x 5 mm VQFN (32)	5 x 5 mm VQFN (40)	7 x 7 mm VQFN (48)
CC1310	X		X								32-128	16-20 + 8	10-30	X	X		X
CC1311R3	X		X								352	32 + 8	22-30			X	X
CC1311P3	X		X							X	352	32 + 8	26				X
CC1312R	X		X	X							352	80 + 8	30				X
CC1312R7	X		X	X	X				X		704	144 + 8	30				X
CC1352R	X	X	X	X		X	X	X	X		352	80 + 8	28				X
CC1352P	X	X	X	X		X	X	X	X	X	352	80 + 8	26				X
CC1352P7	X	X	X	X	X	X	X	X	X	X	704	144 + 8	26				X
CC2640R2F						X					128	20 + 8	10-31	X	X		X
CC2642R						X					352	80 + 8	31				X
CC2642R-Q1						X					352	80 + 8	31				X
CC2651R3		X				X	X				352	32 + 8	23-31			X	X
CC2651P3		X				X	X			X	352	32 + 8	22-26			X	X
CC2652R		X				X	X	X	X		352	80 + 8	31				X
CC2652RB		X				X	X	X	X		352	80 + 8	31				X
CC2652R7		X				X	X	X	X		704	144 + 8	31				X
CC2652P		X				X	X	X	X	X	352	80 + 8	26				X
CC2652P7		X				X	X	X	X	X	704	144 + 8	26				X

Abbildung 3.3: Test des Messagebrokers Mosquitto

Hier in der Tabelle sind die unterstützten Protokolle der einzelnen Modelle sowie deren Leistungsfähigkeit aufgeführt. Spannenderweise ist zu sehen, dass die Top Modelle schon den Standard Thread unterstützen, der vermutlich durch das Projekt "Matter" erheblich an Bedeutung gewinnt.

Texas Instruments bietet als Basis für ZigBee Eigententwicklungen eine Z-Stack Bibliothek. Diese stellt grundlegende Funktionen um das ZigBee Protokoll zu implementieren. Mit Texas Instruments Code Composer Studio steht eine IDE bereit, um den Entwicklungsprozess zu unterstützen.

In dem OpenSource Projekt "zigbee2mqtt" werden ausschließlich Chips von Texas Instruments unterstützt. Es sei erwähnt, dass die meisten gängigen Anbieter von Microchips entsprechende Modelle im Angebot haben.

3.3 Versuchshardware

3.3.1 RaspberryPi

Der RaspberryPi ist ein ARM basierter Computer im Mini-Format. Er dient in diesem Versuch als Server, der die Applikationen hostet und gleichzeitig als Versuchs PC, von dem der Versuch aus ausgeführt wird. Die eingesetzten Dienste sind alle als Webservice implementiert, und daher vollständig auf Kom-

mandozeile parametrierbar, sowie mit WebGUI bedienbar.

Der RaspberryPi besitzt die Nutzer PC typischen Schnittstellen wie Ethernet, HDMI, sowie USB. Als Hauptspeicher wird eine SD-Karte eingesetzt. Dies ist ein erheblicher Vorteil beim vorbereiten mehrerer RaspberryPis für den Versuch.

Auf dem RaspberryPi wird das Linux-basierte Betriebssystem RaspbianOS eingesetzt. Durch die enorme Verbreitung ist hier mit regelmäßigen Updates in Zukunft zu rechnen.

3.3.2 RaspberryPi Zigbee Hat

Als Zigbee Koordinator kommt ein auf dem TI CC2652 basierendem RaspberryPi Hat zum Einsatz. Dieser wird mit einer Firmware aus dem zigbee2Mqtt Repository geflasht.

3.3.3 CC2235 Sniffer Stick

Mit diesem Stick wird die ZigBee Kommunikation zwischen den einzelnen Devices sowie dem Koordinator mitgeschnitten auf einem bestimmten Kanal mitgeschnitten.

3.3.4 Phillips Hue Komponenten

Die Lampen werden in dem Versuch als Demonstrationsobjekte eingesetzt. Sie können Ein- und Ausgeschaltet werden, sowie gedimmt werden. Zusätzlich wird einer Phillips Hue Fernbedienung verwendet, die zur Steuerung der Lampen dient.

3.4 Eingesetzte Software

3.4.1 Raspbian OS

RaspbianOS ist eine Linux Distribution, welche direkt vom Hersteller des RaspberryPis speziell auf die Bedürfnisse des Board angepasst werden. Es enthält eine Desktop Umgebung sowie die Grundlegend wichtigen Paketen. Es basiert auf Debian, damit sind auch die entsprechenden Paketquellen verfügbar.

3.4.2 Docker

Docker ist eine Container Umgebung, um Anwendungen containerisiert auf Linux-Servern laufen lassen zu können. Docker reduziert erheblich den Aufwand, Anwendungen auf mehreren Server auszurollen. Alle Abhängigkeiten sind im Container enthalten, sodass hier keine Komplikationen mit anderen Anwendungen zu befürchten sind.

3.4.3 zigbee2mqtt

zigbee2mqtt ist ein offenes Softwareprojekt auf Zigbee, welches aus mehreren Komponenten besteht.

TI CC Firmware

Firmware für die Texas Instruments Chips, um diese als Koordinator einsetzen zu können. Die Firmware basiert auf dem Z-Stack von Texas Instruments.

zigbee-herdman

Dieses Modul verbindet sich direkt mit dem Zigbee Adapter, und steuert ihn über die TI zStack monitoring and test API. [Ins12]

zigbee-herdman-converters

Dieser Konverter kann proprietäre Cluster die durch Geräte exposed werden umwandeln in standard Cluster. Mit diesem Converter lassen sich sämtliche Geräte so adaptieren, dass sie nach Wunsch gesteuert und ausgelesen werden können.

zigbee2mqtt

Das Hauptmodul stellt die WebGui sowie eine Webanwendung mit einer SQLite Datenbank. Die Webanwendung und die Datenbank verwalten den Zustand des Netzwerkes und die angebundenen Geräte. Die WebGUI dient zur Administration des Koordinators.

1. Bild von zigbee2mqtt

Die WebGUI enthält eine große Anzahl von Funktionen, die weitaus tiefer reichen als für die Nutzung notwendig sind. Prinzipiell sind die meisten ZigBee Geräte direkt Einsatzfähig, wenn ein Community Mitglied dieses bereits in der Anwendung angelegt hat. Es ist auch möglich, eigene Beschreibungen für nicht unterstützte Geräte zu erstellen.

3.4.4 Wireshark

Wireshark ist eine quelloffene Anwendung um Datenströme mitzuschneiden und zu untersuchen. Es kann durch Verwendung von Packetsniffern wie nPcap verschiedenste Medien wie zum Beispiel Ethernet und USB mit entsprechenden Protokollen verarbeiten.

Kapitel 4

Versuchsdurchführung

In diesem Kapitel wird der Hardware- und der Softwareaufbau des Versuches beschrieben.

4.1 Versuchsaufbau

Der Student erhält folgende Komponenten:

- RaspberryPi 3
- CC2531 Sniffer Stick
- cod.m ZigBee CC2652P2 Raspberry Pi Module
- 2 x Phillips Hue White E27
- 1 x Phillips Hue dimmer switch
- HDMI Kabel
- Ethernet Kabel

Der Student muss KVM Komponenten selbst bereitstellen. Alternativ, wenn der Versuch in der Hochschule durchgeführt werden sollte, können diese auch durch die Hochschule gestellt werden. Der ZigBee Koordinator ist als “Hat “ auf dem Raspberry installiert. Der Sniffer ist per USB an der Frontseite des Raspberrys angeschlossen.

4.2 Versuchssoftware

Folgende Software wird eingesetzt:

- RaspbianOS

- Docker
- zigbee2mqtt
- mosquitto

Auf einem RaspberryPi werden die Anwendungen zigbee2mqtt, Mosquitto sowie HomeAssistant per Docker ausgeführt. Die Services sind konfiguriert, es sind keine Geräte per Zigbee verbunden. Die jeweiligen Webinterfaces sind über eine Webadresse im Browser erreichbar. Der ZigBee Koordinator ist als “Hat “ auf dem Raspberry installiert. Der Sniffer ist per USB an der Frontseite des Raspberrys angeschlossen.

Das Deployment des Versuchs wird im Kapitel LCM beschrieben.

4.3 Aufgabenstellungen

In diesem Kapitel werden die Aufgabenstellungen aus dem LabGuide genauer beschrieben, sowie eine Musterlösung gegeben.

4.3.1 Aufgabe 1 - Vorbereitungen

Vorbereitung

Schließen sie an den RaspberryPi einen Monitor, Tastatur, Maus sowie den Sniffer Stick an. Durch Anschluss der Stromversorgung startet der Raspberry automatisch. Melden sie nach starten des Betriebssystems mit folgenden Zugangsdaten an:

- User: Student
- Password: ZigbeeLab

Starten sie ein Konsolenfenster und überprüfen mit folgendem Befehl, ob die benötigten Container ausgeführt werden:

```
1 | > docker ps
```

Es sollten 3 Container im Status “Running“ sein. Beschreiben sie in eigenen Worten welche Container sie hier sehen.

ZigBee2Mqtt Einrichtung

Starten sie den Webbrowser Firefox und besuchen die Webseite:

```
1 | https://zigbee2mqtt.local
```

Überprüfen, dass keine Geräte mit dem Koordinator verbunden sind. Im Zweifelsfall können sie den Versuch zurücksetzen. Dies wird in den FAQs beschrieben.

Stellen sie den Kanal, den der Zigbee Koordinator nutzen soll nun auf den durch Ihren Professor vorgegeben Wert. Dies verhindert, dass sich die Studenten gegenseitig beeinflussen. Zuhause können sie diesen Schritt überspringen.

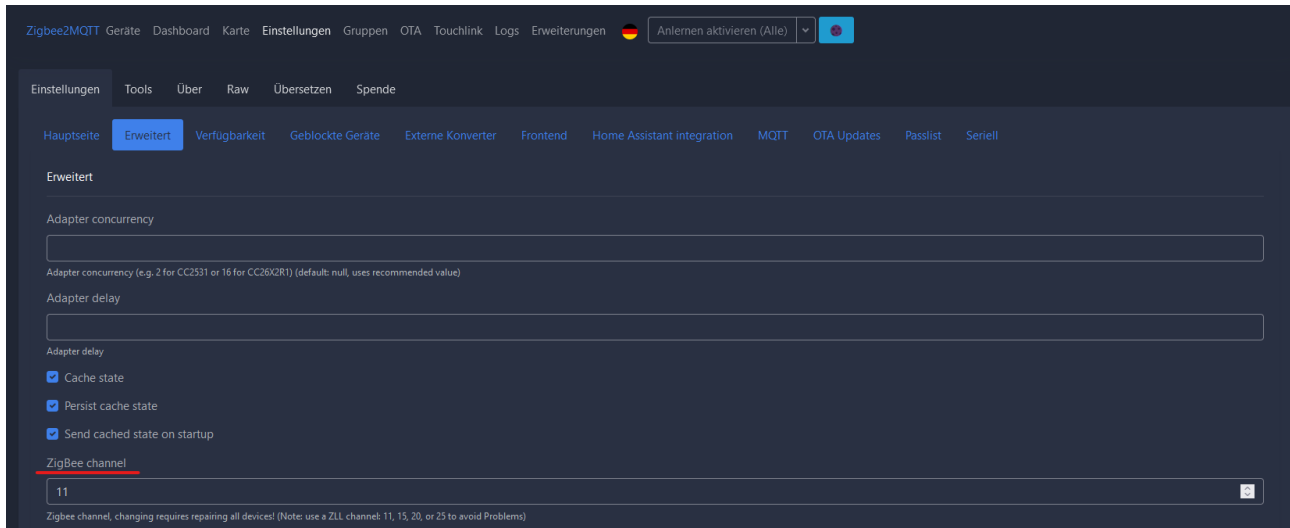


Abbildung 4.1: Zigbee Kanal Einstellung

Per Default verwendet Zigbee2Mqtt einen zufällig gewählten Netzwerkschlüssel. In dem Versuch wird der Schlüssel aber vorgegeben, um die Pakete entschlüsseln zu können. Bitte setzen sie folgenden Schlüssel:

```
1 | 0x 00 00 ... 00 <Gruppe>
```

Die Einstellung finden sie unter der Kanal Einstellung als “Network key(string)“.

Achten sie darauf, im Anschluss die Einstellung am Ende der Seite zu bestätigen. Dafür klicken sie auf den “Submit “ Button am Ende der Seite

Wireshark Einrichtung

Mit folgendem Befehl können sie ein Wireshark Capture auf entsprechenden Kanal starten: Starten sie ein Konsolenfenster und testen sie, ob der Befehl erfolgreich ausgeführt wird. Wireshark sollte starten, und Pakete sollten ersichtlich sein. Für “<Kanal> “ setzen sie den von Ihnen gewählten Kanal ein.

```
1 | > zbwireshark -c <Kanal>
```

Beenden sie den automatisch gestarteten Capture Vorgang. Gehen sie in das Menü: Bearbeiten > Einstellungen > Protokolle > ZigBee > Edit (Pre-configured Keys) und tragen hier den “TC-Link Key“ und den “Network Key“ ein. Als “Network Key“ verwenden sie den in Zigbee2Mqtt gesetzten Key. Der

“TC-Link Key“ ist ein Standard-Key, der verwendet werden muss.

1 | 0x 5A 69 67 42 65 65 41 6C 6C 69 61 6E 63 65 30 39 (ZigBeeAlliance09)

Manche Hersteller verwenden proprietäre Schlüssel. Die Geräte sind dann nicht mit Koordinatoren anderer Hersteller kompatibel.

Hinweis

Alle Aufgaben sollen mit Wireshark mitgeschnitten werden. Lesen sie die Aufgabenstellung erst durch und machen sie sich den Ablauf klar. Versuchen sie das Zeitfenster des Wireshark Mitschnitts so kurz wie möglich zu halten, und in dieser Zeit nur die in der Aufgabenstellung beschriebenen Aktionen durchzuführen. Anderenfalls wird Ihr Mitschnitt sehr unübersichtlich.

4.3.2 Aufgabe 2 - Joining einer Phillips Hue Lampe

Durchführung

Schalten sie eine der beiden Zigbee Lampen ein. Die Lampe sollte leuchten. Dies ist das Standardverhalten, wenn die Lampen in keinem ZigBee Netz integriert sind. Starten sie nun ein Wireshark Capture und erlauben in Zigbee2Mqtt das Anlernen von Geräten. Sobald Zigbee2Mqtt ein erfolgreiches Interview gemeldet hat, beenden sie den Capture Vorgang. Die Lampe signalisiert durch ein kurzes blinken ebenfalls einen erfolgreichen Interview.

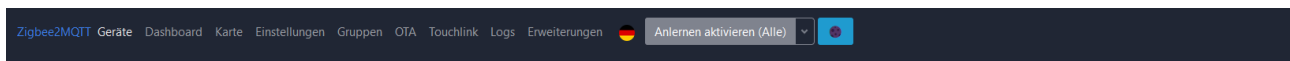


Abbildung 4.2: Zigbee Anlernen aktivieren

Hinweis

Speichern sie den Wireshark Capture ab als “<Gruppe> - ZigbeeLab - Aufgabe 2.1“

Navigieren sie nun zur Übersichtsseite der Lampe. Diese sollte ähnlich wie folgende Seite aussehen:

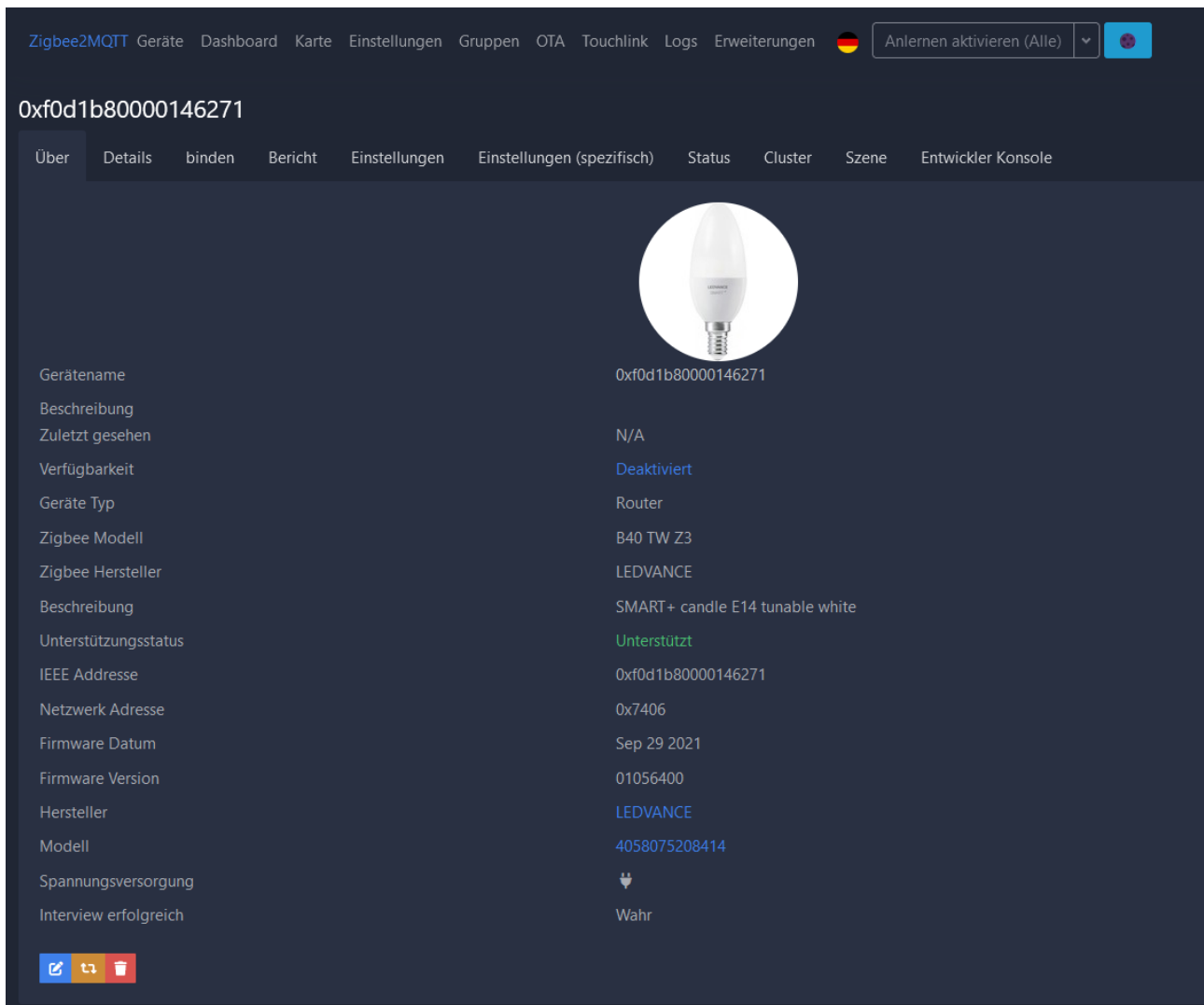


Abbildung 4.3: Zigbee Device Übersicht

Vergeben sie in der Übersichtsseite der Lampe einen Nutzerfreundlichen namen. Die geschieht über den blauen Button im unteren Teil der Übersicht. Dimmen und schalten sie die Lampe über die Weboberfläche. Die ist unter dem Reiter “Details“ möglich. Starten sie einen weiteren Capture Vorgang, in dem sie einen Schaltvorgang mitschneiden.

Hinweis

Speichern sie den Wireshark Capture ab als “<Gruppe> - ZigbeeLab - Aufgabe 2.2“

Fragen

Hinweise

Sollte die Lampe sich nicht Verbinden, kann es notwendig sein die Lampe zurückzusetzen. In den FAQs finden sie zwei Methoden dazu.

4.3.3 Aufgabe 3 - Joining einer Fernbedienung über die Lampe

4.3.4 Durchführung

Für diese Aufgabe sollte nur die eine Lampe mit dem Koordinator verbunden sein. Die Fernbedienung wird nun über die Lampe dem Netzwerk hinzugefügt. Aus diesem Grund wird es nur der Lampe erlaubt, ein neues Gerät aufzunehmen. Drücken sie den Setup Button auf der Fernbedienung, bis die LED dauerhaft grün leuchtet. Ein erfolgreiches anlernen wird auch hier in der Weboberfläche und durch ein blinken der grünen LED signalisiert. Schneiden sie diesen Vorgang wieder mit Wireshark mit und speichern den Capture unter “<Gruppe> - ZigbeeLab - Aufgabe3”.

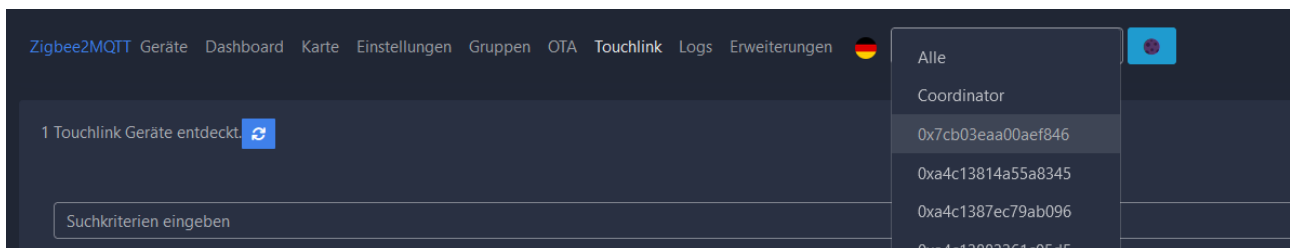


Abbildung 4.4: Zigbee Anlernen aktivieren - nur Lampe

Aufgabe

Speichern sie den Wireshark Capture ab als “<Gruppe> - ZigbeeLab - Aufgabe 3”

Fragen

Hinweise

4.3.5 Aufgabe 4 - Binding der Fernbedienung

4.3.6 Durchführung

Navigieren sie in der Weboberfläche zu der Übersicht Ihrer Lampe. Dort finden sie einen Reiter “binden”. Hier binden sie den Endpunkt X Ihrer Lampe mit dem Endpunkt X Ihrer Fernbedienung. Schalten sie nun die Lampe mit der Fernbedienung ein und aus.

Hinweis

Speichern sie den Wireshark Capture ab als “<Gruppe> - ZigbeeLab - Aufgabe 4”

4.3.7 Aufgabe 5 - Gruppenbildung

Hinweis

Speichern sie den Wireshark Capture ab als “<Gruppe> - ZigbeeLab - Aufgabe 5“

Kapitel 5

Life Cycle Management

In diesem Kapitel geht es um die Pflege, die Bereitstellung sowie die Zurücksetzung des Praktikumversuchs.

5.0.1 Deployment

5.0.2 Zurücksetzen des Versuchs

5.0.3 Update der eingesetzten Software

Abbildungsverzeichnis

Literatur

- [Ama12] Amazon. *Understand Smarthome Zigbee Support*. [Online; Stand 03. Oktober 2022]. 2012.
URL: <https://developer.amazon.com/en-US/docs/alexa/smarthome/zigbee-support.html>.
- [Ins12] Texas Instruments. *Z-Stack Monitor and Test API*. [Online; Stand 05. Oktober 2017]. 2012.
URL: <https://github.com/koenkk/zigbee-herdsman/raw/master/docs/Z-Stack%20Monitor%20and%20Test%20API.pdf>.