Hochschule RheinMain
Fachbereich ITE
Studiengang EE-CS

Scientific Project

Entwicklung eines Zigbee Praktikums auf Grundlage latexmk von FOS Software

verfasst von **Benedikt HEUSER**
Matrikelnummer 105320

am 25.04.2022

Kompiliert am 19.10.2022 um 15:00:56 - Erstellt mit LaTeX

Inhaltsverzeichnis

Kapitel 1

Einführung

In diesem Versuch soll ein Praktikumsversuch für die Vorlesung Internet of Things für Professor Jürgen Winter entwickelt werden. In dem Versuch soll das Verhalten des ZigBee Protokolles erforscht werden. Den Studenten soll ein Raspberry Pi sowie diverse Zigbee Geräte und Adapter ausgehändigt werden. Damit kann ein ZigBee Netzwerk aufgebaut, und anschließend mit dem Sniffer Tool Wireshark analysiert werden.

1.1 Anforderungen an die Prakukumsarbeit

Die Anforderungen an der Versuch werden an dieser Stelle definiert und mit Indexiert, um im weiteren Verlauf darauf Bezug nehmen zu können.

- **A010** - Der Versuch soll an einem Tag durchführbar sein.
- **A020** - Der Versuch soll kein Vorwissen in Linux vorraussetzen
- **A030** - De Versuch setzt Vorwissen in Paketorientierten Datenübetragung vorraus.
- **A040** - Der Versuch setzt Vorwissen in der Bedienung von Wireshark vorraus.
- **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-Netwerkes 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 Batteriebetriene 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 entscheidentes 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 addresierbar.

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 “deCONZ “. Als komplette Produktlinie für den nicht technisch visierten Endkungenmarkt gibt es die Produktsparte "Phoscon", hautpsächlich 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 Kompatiblität 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 Kompatiblität 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 Kompatiblität 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 unterstüztzter 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 sollen alle verwendeten Komponenten und Technologien kurz erklärt werden.

3.1 ZigBee

Die ZigBee Alliance wurde durch die Hersteller 1.Zigbee Gründer gegründet, um einen einheitlichen Übertragungstandard voranzubringen. ZigBee basiert zwar auf einem offenem IEEE Standard, bringt aber Lizenzpflichtige Komponenten mit. Dies verhindert leider maßgeblich eine weitere Ausbreitung des Standards.

ZigBee ist ein Kommunikationsprotokoll, welches im Bereich Iot Anwendungen findet. Das Protokoll baut auf dem Standard 802.40 auf. Genutzt wird es, um IoT fähige Geräte in einem Haushalt, wie zum Beispiel Lampen, Schalter oder Thermostate zur Kommunikation zu befähigen. Markantetes Merkmal am Protokoll ist, dass die Geräte keine direkte Funkverbindung zu einem zentralen Controlle brauchen, sondern über andere ZigBee fähige Geräte ein Netzwerk aufbauen. Vorteil ist, dass ein im Vergleich zur benötigten Sendeleistung sehr großer Radius und Anahl von Geräten abgedeckt werden kann.

ZigBee erweitert den IEEE Standard um 2.Zigbee erweiterungen

3.2 Texas Instruments CC Chips

Texas Instruments bietet ein großes Spektrum von Microcontrollern, die den Zigbee Standard beherschen. Die Chips lassen sich in SDK Kits erwerben um eigene Firmwares zu entwickeln. Ebenfalls lassen sich im Internet sehr günstige USB-Donge erwerben, welche sich nach belieben Flashen lassen. Dies erspart eine sehr aufwendige PCB Entwicklung oder das erwerben des kostenspieligen SDK Kits von Texas Instruments.

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 3.1: 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 Unterstüzung 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 3.2: 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 "Matterërheblich 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 TexasInstruments Code

Composer Studio steht eine IDE bereit, um den Entwicklungsprozess zu unterstützen.

In dem OpenSource Projekt ßigbee2mqtt"werden ausschließlich Chips von Texas Instruments unterstützt. Es sei erwähnt, dass die meißten 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 Kommandozeile parametrierbar, sowie mit WebGUI bedienbar.

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

Auf dem RaspberryPi wird das Linux-basierte Betriebssystem RaspbyanOS eingesetzt. Durch die enorme Verbreitung ist hier mit regelmäígen 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 dein 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 Hie 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 enthällt 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.

3. Bild von zigbee2mqtt

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

3.4.4 Wireshark

Wireshark ist eine quelloffene Anwendung um Datenstöme 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

Versuchsaufbau

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

Kapitel 5

Versuchsdurchführung

5.1 Ursprungszustand vor Versuch

Der ZigBee Kordinator ist als “Hat “ auf dem Raspberry installiert. Der Sniffer ist per USB an der Frontseite des Raspberrys angeracht. Eingabegeräte, ein Monitor sowie die Stromversorgung sind weiterhin vom Studenten anzuschließen.

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

5.2 Aufgabenstellungen

5.2.1 Aufgabe 1 - Vertraut machen mit der Umgebung

Der Student soll den Webbrowser starten, und z2m.local aufrufen. Er soll ich Anhand einer kurzen Beschreibung selbst mit der WebGui vertraut machen. Der Koordinator soll auf den entsprechenden Kanal eingestellt werden.

Der Student soll sichergehen, dass keine Geräte in zigbee2mqtt registriert sind. Falls dies der Fall ist, soll er das Script zum zurücksetzen des Versuches starten.

Der Student soll Wireshark über die Kommandozeile mit den entsprechendem Befehl starten, und überprüfen ob die Anwendung ohne Fehlermeldung startet und orgnungsgemäß funktioniert.

5.2.2 Aufgabe 2 - Joining der Phillips Hue Lampe und Fernbediernung

Der Student soll das Beitreten von Komponenten in zigbee2mqtt erlauben. Erst anschließend soll er Wireshark starten. Nun lässt er die Phillips Hue Lampe sowie die Fernbedienung dem Netzwerk

beitreten. Sobald zigbee2Mqtt ein erfolgreiches Interview meldet, wird Wireshark gestoppt, und der Sniffing Vorgange abgespeichert.

Kapitel 6

Life Cycle Management

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

6.0.1 Deployment

6.0.2 Zurücksetzen des Versuchs

6.0.3 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.