Absicherung eines Röntgenraumes

Verknüpfung von ZigBee und MQTT

Dominik Cremer, Dominic Inger, Nicols Bloß

2024

Inhaltsverzeichnis

[Einleitung 2](#_Toc169089589)

[Hardware-Komponenten 2](#_Toc169089590)

[Installation von Software-Komponenten 2](#_Toc169089591)

[Phoscon 2](#_Toc169089592)

[Anschließen 2](#_Toc169089593)

[Installation 2](#_Toc169089594)

[Konflikte und Probleme: 4](#_Toc169089595)

[Bewertung 4](#_Toc169089596)

[ZigBee2MQTT 4](#_Toc169089597)

[Schaltungsplan 5](#_Toc169089598)

[Programmcode 6](#_Toc169089599)

[Quellen 6](#_Toc169089600)

# Einleitung

Diese technische Dokumentation beschreibt die Planung, Installation und Inbetriebnahme eines Sicherheitssystems für einen Röntgenraum. Ziel dieses Projekts ist es, die Sicherheit während der Nutzung des Röntgengeräts in Raum R zu gewährleisten und ein versehentliches Betreten des Raums während einer Röntgenaufnahme zu verhindern. Dazu soll ein umfassendes System aus Sensoren und Signalleuchten installiert werden, das sowohl den Status der Türen anzeigt als auch im Notfall die Röntgenaufnahme automatisch stoppt.

# Hardware-Komponenten

* Raspberry Pi inkl. Strom- und Monitorkabel
* GPIO-Board
* Aqara Door/Window Proximity Sensor
* Tradfri Glühlampe
* ConBee II ZigBee USB-Gateway
* LEDs
* Knopf
* Widerstände und Stromkabel
* Maus und Tastatur

# Installation von Software-Komponenten

## Phoscon

### Anschließen

Für den optimalen Betrieb wird die Nutzung eines USB-Verlängerungs­kabels empfohlen.

* Verhindert Interferenzen mit Gehäuse und Peripherie
* maximale Signal­reichweite für zuverlässige Steuerung

### Installation

1. USB-Zugriffsrechte für Nutzer setzen  
    *sudo gpasswd -a $USER dialout*
   * fügt den aktuellen Benutzer ($USER) zur "dialout"-Gruppe hinzu.

1. --> Die Zugriffsrechte werden erst nach dem Aus- und Einloggen bzw. nach einem Neustart aktiv.

1. Phoscon Public-Key importieren  
   *wget -qO-* [*https://phoscon.de/apt/deconz.pub.key*](https://phoscon.de/apt/deconz.pub.key) *| gpg --dearmor | \  
    sudo tee /etc/apt/trusted.gpg.d/deconz-keyring.gpg >/dev/null*
   * lädt den öffentlichen GPG-Schlüssel von "phoscon.de/apt/deconz.pub.key" herunter, entschlüsselt ihn mit GPG und leitet die Ausgabe an "sudo tee /etc/apt/trusted.gpg.d/deconz-keyring.gpg" um, um den Schlüssel im GPG-Format zu speichern. Dabei wird die Standardeingabe (wget) und die Standardausgabe (tee) umgeleitet, um keine unnötige Ausgabe zu erzeugen (>/dev/null).

1. APT-Repository für deCONZ konfigurieren  
   **Stable***sudo sh -c "echo 'deb* [*http://phoscon.de/apt/deconz*](http://phoscon.de/apt/deconz) *\  
    generic main' > \  
    /etc/apt/sources.list.d/deconz.list"*

Dieser Befehl erstellt eine neue Datei namens "deconz.list" im Verzeichnis "/etc/apt/sources.list.d/" und fügt die Zeile "deb <http://phoscon.de/apt/deconz> generic main" hinzu. Dabei wird "sudo sh -c" verwendet, um die Ausführung des Befehls mit Root-Rechten zu ermöglichen.

1. APT-Paketliste aktualisieren  
    *sudo apt update*
   * aktualisiert die Paketlisten des Paketmanagers

1. deCONZ installieren  
    *sudo apt install deconz*

Nach der Installation kann deCONZ über das Applikations­menü gestartet werden.

*Menu > Programming > deCONZ*

* Anschließend obene rechts das Web-Interface öffnen, nach dem Gateway suchen und anmelden.

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

### Konflikte und Probleme:

#### Phoscon:

* Nach Installation der Phoscon Software, gab es keine weiteren Anweisungen für die Kopplung von Geräten

--> Web-Interface öffnen

--> erfordert einen Neustart der Software, sodass das Gateway gefunden werden kann

#### IKEA-Lampe:

* In der Anleitung wurde nicht angegeben, wie man das Gerät ohne Fernbedienung zurücksetzt.

--> Ein-/Ausschalter 6 mal betätigen, sodass die Lampe blinkt

* Nach dem Koppeln über das Phoscon Webinterface, wird das Gerät auch in der Desktop-App angezeigt, jedoch wird die Verbindung nach kurzer Zeit aus unerklärlichen Gründen unterbrochen.

#### Magnetsensor:

* Kopplung erfolgt durch die Reset-Taste (3s), so wie es auch in der Anleitung beschrieben ist.

--> Nach erfolgreichen Koppeln wird der Status noch für etwa 10 Sekunden auf der Phoscon Seite aktualisiert. Verliert dann aber ebenfalls die Verbindung.

#### Interferenzen:

* Überlappung von Signalen (2,4GHz) durch verschiedene WLAN Signale und ZigBee-Geräte
* Geräte von anderen Gruppen werden gefunden und konfiguriert, obwohl dies nicht die Intention ist

### Bewertung

Bei dem Einrichten der Software Phoscon sind keine Probleme aufgetreten und auch die Anleitung war klar und verständlich. Allerdings hat die Einrichtung von ZigBee Geräten nicht einwandfrei funktioniert, da unter anderem auch die Anleitungen Teils mangelhaft waren und einige Schritte ausgelassen wurden. Des Weiteren trat das Problem auf, dass sich die Signale von WLAN und ZigBee der verschiedenen Geräte und Gruppen überlappen, sodass die Verbindung entweder gestört wurde oder teils Geräte anderer Gruppen angezeigt wurden und konfiguriert werden konnten.

Da es sich um ein Smart-Home-System handelt, ist die Datensicherheit von entscheidender Bedeutung. Letzterer Punkt des oberen Absatzes ist somit teils eher negativ anzusehen. Es müssen gegeben falls zusätzliche Sicherheitsmaßnahmen gemäß den verwendeten Geräten und der Angaben von Geräteherstellern vorgenommen werden, sodass sensible Informationen nicht gefährdet sind. Dies ist bei einer Lampe zwar nicht unbedingt das größte Sicherheitsbedenken, jedoch sollte man dennoch Sicherheitsmaßnahmen in Betracht ziehen.

## ZigBee2MQTT

# Lokalisieren der Position des ConBee-Adapters

ls -l /dev/serial/by-id

# Nodesource-Repositories hinzugefügen und Node.js sowie andere benötigte Pakete installieren

sudo curl -fsSL https://deb.nodesource.com/setup\_20.x | sudo -E bash -

sudo apt-get install -y nodejs git make g++ gcc libsystemd-dev make

# Überprüfung der Installierten Version von "NodeJS und "NPM", um spätere Komplikationen zu vermeiden

node --version

npm --version

# Verzeichnis für ZigBee2MQTT erstellen und den aktuellen User als Besitzer berechtigen

sudo mkdir /opt/zigbee2mqtt

sudo chown -R ${USER}: /opt/zigbee2mqtt

# Das ZigBee2MQTT-Repository wird herunterladen

git clone --depth 1 https://github.com/Koenkk/zigbee2mqtt.git /opt/zigbee2mqtt

# Hier wird in das ZigBee2MQTT-Verzeichnis gewechselt und die Abhängigkeiten installiert

cd /opt/zigbee2mqtt

npm ci

# Aufbauen der NPM-App

npm run build

# Konfigurationsdatei in zuvor erstelltes Verzeichnis für ZigBee2MQTT kopieren

cp /opt/zigbee2mqtt/data/configuration.example.yaml /opt/zigbee2mqtt/data/configuration.yaml

# Konfigurationsdatei auf das Netzwerk und den verwendeten Adapter abändern (falls nötig)

nano /opt/zigbee2mqtt/data/configuration.yaml

# In das korrekte Verzeichnis wechseln, sodass diese Daten verwendet werden können

cd /opt/zigbee2mqtt

# NPM starten (kann über den Browser mit dem Port 8080 zugegriffen werden)

npm start

# Schaltungsplan

Ein Bild, das Diagramm, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

# Programmcode

# Quellen

* + <https://www.zigbee2mqtt.io/guide/installation/01_linux.html#optional-running-as-a-daemon-with-systemctl>
  + <https://www.zigbee2mqtt.io/guide/installation/01_linux.html#determine-location-of-the-adapter-and-checking-user-permissions>