

Punkte Dokumentation:

- Pflichtenheft: 4,5/5

- QS-Plan: 4,0/5



**Hochschule für Technik  
und Wirtschaft Berlin**

*University of Applied Sciences*

Technische Spezifikation  
im Fachübergreifendes Projekt

Sprachsteuerung eines Hauses

Autoren ← : Azim Izzudin Ramadhani Mubarak  
Bashar Mustafa  
Kenneth Austin  
Reynaldo Domenico

Betreuer/in : Prof. Dr.-Ing. Christian Müller

Ort, Datum : Berlin, 14.06.2022

## Inhaltsverzeichnis

Abbildungsverzeichnis .....	II
Tabellenverzeichnis .....	III
Verzeichnis vorhandener Dokumente .....	V
<b>1 Prozessüberblick .....</b>	<b>1</b>
1.1 Smarthome Workflow .....	1
1.2 Sprachbefehl Workflow .....	2
1.3 Update Workflow .....	3
<b>2 Technische Spezifikation SW .....</b>	<b>4</b>
2.1 Überblick Komponenten .....	4
2.2 Beschreibung der Implementierung .....	5
2.2.1 OpenHABian auf Raspberry PI installieren .....	5
2.2.2 MQTT .....	6
2.2.3 Geräte per MQTT verbinden .....	7
2.3 System Infrastruktur .....	8
<b>3 Technische Spezifikation Konstruktion .....</b>	<b>9</b>
3.1 Baugruppen .....	9
<b>4 Modul Abhängigkeiten .....</b>	<b>10</b>

kein Unterkapitel 3.1, wenn kein Unterkapitel 3.2 folgt.

## **Abbildungsverzeichnis**

Abbildung 1: Der Smarthome Workflow .....	1
Abbildung 2: Der Sprachbefehl Workflow .....	2
Abbildung 3: Der Update Workflow .....	3
Abbildung 4: Komponentendiagramm .....	4
Abbildung 5: PuTTY konfigurieren.....	5
Abbildung 6: Flash via Etcher .....	6
Abbildung 7: Gerät wird mit Tuya-Convert konfiguriert .....	7
Abbildung 8: System Infrastruktur .....	8
Abbildung 9: Technische Zeichnung .....	9

## Tabellenverzeichnis

Tabelle 1: Version Historie.....	IV
Tabelle 2: Relevante Dokumente .....	V
Tabelle 3: Komponente und Funktionen Verbindung .....	4
Tabelle 4: OpenHABian auf Raspberry PI Tabelle .....	5
Tabelle 5: MQTT Tabelle .....	6
Tabelle 6: Geräte per MQTT verbinden.....	7
Tabelle 7: Modul Abhängigkeiten .....	10

## Copyright

© Sprachsteuerung eines Hauses

Die Weitergabe, Vervielfältigung oder anderweitige Nutzung dieses Dokumentes oder Teile davon ist unabhängig vom Zweck oder in welcher Form untersagt, es sei denn, die Rechteinhaber/In hat ihre ausdrückliche schriftliche Genehmigung erteilt.

## Version Historie

*Tabelle 1: Version Historie*

Version	Datum	Verantwortlich	Änderung
0.1	07.06.2022	Kenneth	Initiale Dokumenterstellung
0.2	08.06.2022	Alle	Überblick und Workflow hinzugefügt
0.3	12.06.2022	Alle	Technische Spezifikation SW hinzugefügt
0.4	12.06.2022	Alle	Abbildung und Tabelle hinzugefügt
0.5	13.06.2022	Alle	Erweiterungen
0.6	14.06.2022	Kenneth	Letzte Überprüfung
<b>1.0</b>	<b>14.06.2022</b>	<b>Kenneth</b>	<b>Abgabe</b>

## Verzeichnis vorhandener Dokumente

Alle für die vorliegende Spezifikation ergänzenden Unterlagen müssen hier aufgeführt werden.

*Tabelle 2: Relevante Dokumente*

Dokument	Autor	Datum
Lastenheft-Gruppe-4.pdf	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	26.04.2022
Projektplanung.mpp	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	24.05.2022
Pflichtenheft-Gruppe-4.pdf	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	24.05.2022
Qualitätsicherung-Gruppe-4.pdf	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	14.06.2022

## 1 Prozessüberblick

In diesem Dokument wird das Design des Projekts "Sprachsteuerung eines Hauses" der Fachübergreifenden Projekte im SoSe 22 an der HTW Berlin spezifiziert.

Es wird ein Gerät entwickelt, das den Zugriff auf verschiedene "Smart Devices" nur mit Sprachsteuerung ermöglicht. Dieses Gerät besteht aus Software und Hardware. Für die Hardware wird es mit Komponenten hergestellt, die aus Mikrofon, Lautsprecher, Raspberry Pi, Gehäuse und LED bestehen. Die LED leuchtet auf, wenn ein Wort oder Code als „Wake Word“ genannt, vom Benutzer gesagt wird. Das zeigt an, dass unser Gerät bereit ist, Befehle abzuheören. Das Mikrofon wird verwendet, um die Stimme des Benutzers zu erfassen und an den Mikrocontroller weiterzuleiten.

Der Raspberry Pi funktioniert hier als Mikrocontroller, in den wir alle Programme und Funktionen schreiben werden. Der Lautsprecher dient dazu, Feedback zu geben, dass der vom Lautsprecher erfasste Befehl ausgeführt werden kann oder nicht. Wenn der vom Benutzer gegebene Befehl erfolgreich ist, leuchten die gewünschten "Smart Devices" auf und handeln gemäß dem Befehl.

Um alle Komponenten abzudecken, verwenden wir ein Gehäuse, das mit 3D-Drucker gedruckt wird. Für die Software wird dieses Gerät, Rhasppy als Spracherkennung und openHab zum Implementieren des Befehls verwenden. Um die Verbindung zwischen Geräte und openHAB zu erschaffen wird mit MQTT Protokoll verwendet für die „Bindings“- Verbindung.

### 1.1 Smarthome Workflow

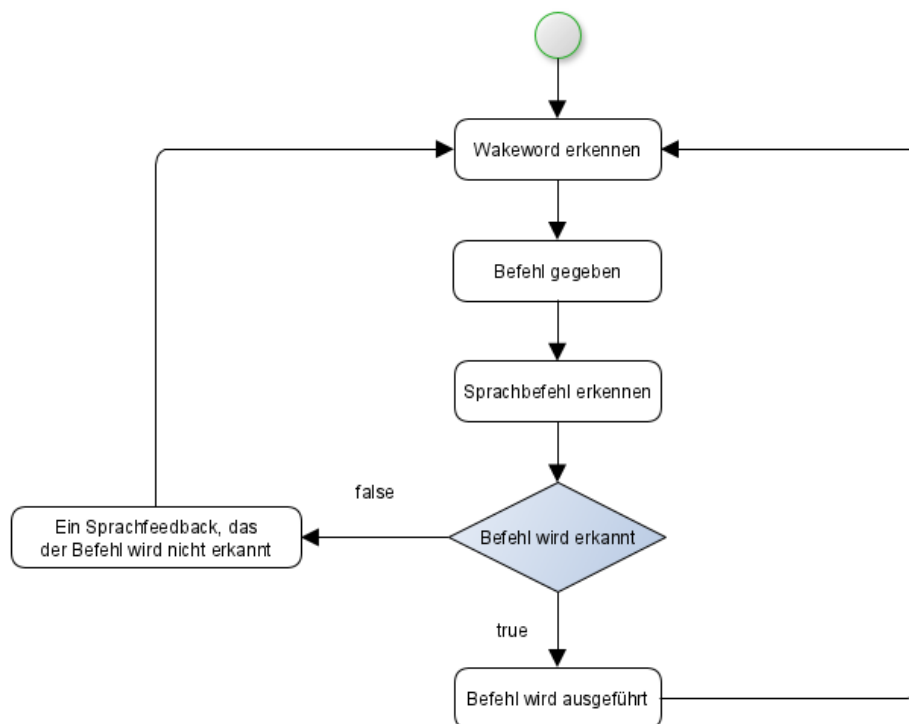


Abbildung 1: Der Smarthome Workflow

## 1.2 Sprachbefehl Workflow

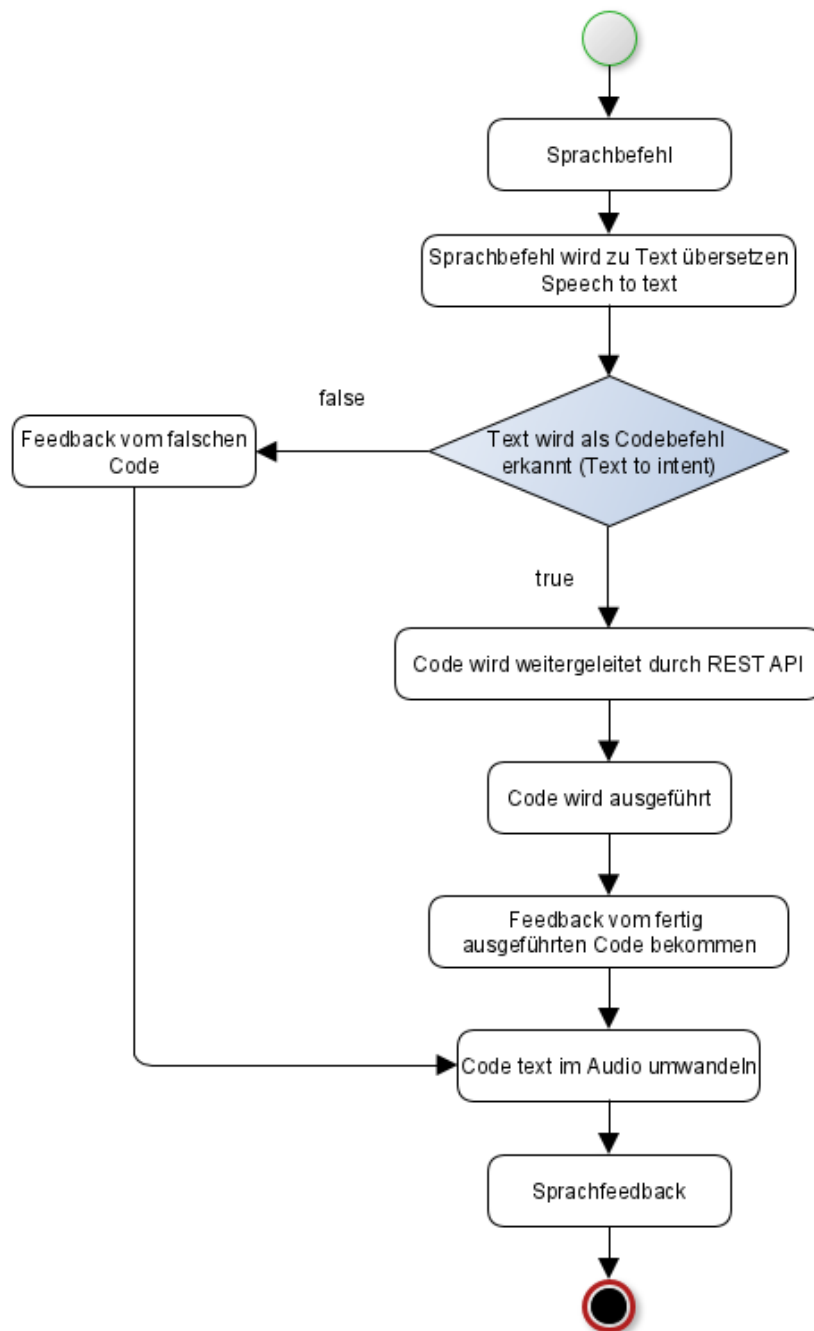


Abbildung 2: Der Sprachbefehl Workflow



### 1.3 Update Workflow

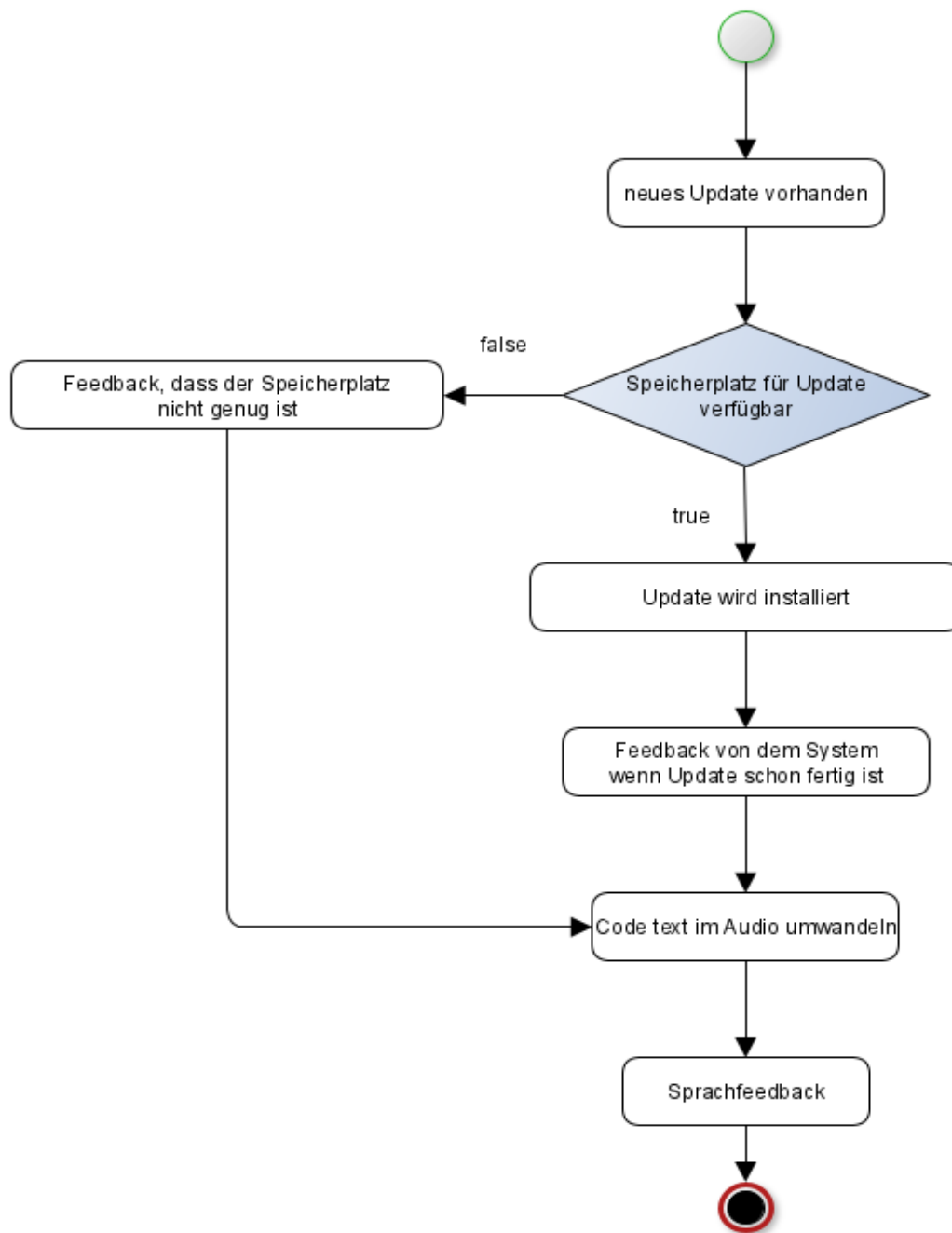


Abbildung 3: Der Update Workflow

## 2 Technische Spezifikation SW

### 2.1 Überblick Komponenten

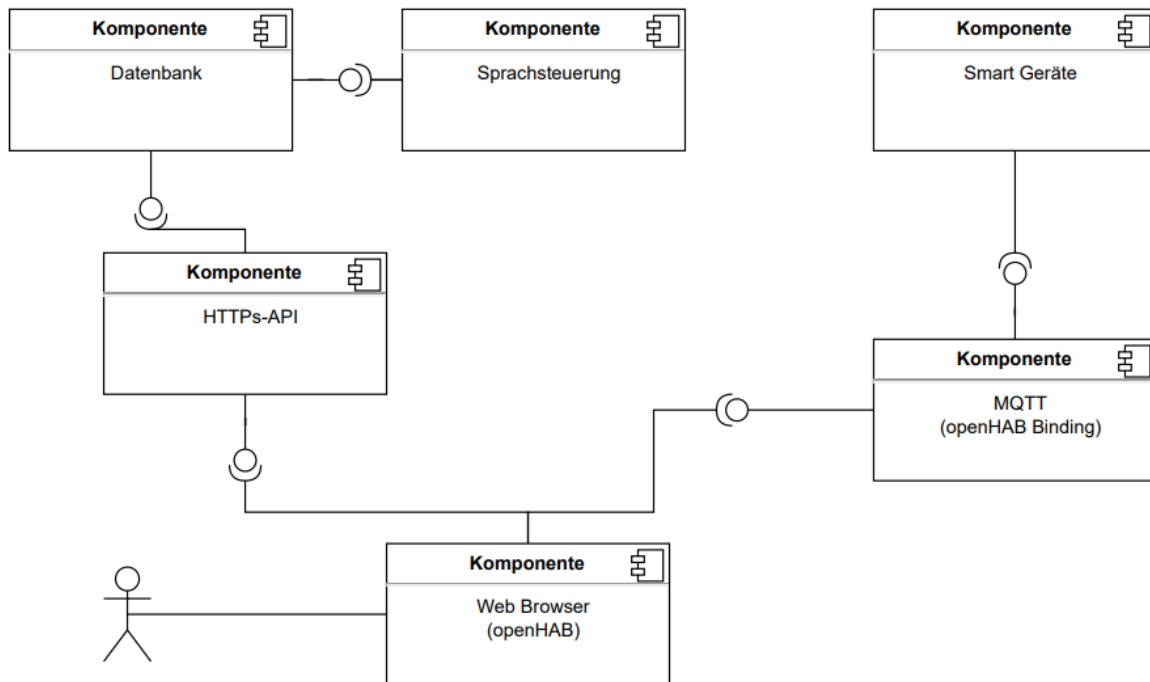


Abbildung 4: Komponentendiagramm

Tabelle 3: Komponente und Funktionen Verbindung

SW-Komponente	Erfasste Funktion aus dem Pflichtenheft
Web-Browser	F1: Befehl geben F6: Lautstärke anpassen
MQTT	F5: Feedback geben F7: Update
HTTPS-API	F7: Update F9: Haushaltgeräte verbinden
Datenbank	F2: Audio im Text umwandeln F4: Text im Audio umwandeln
Smart Geräte	F7: Update F8: WLAN verbinden
Sprachsteuerung	F2: Audio im Text umwandeln F3: Spracherkennung F4: Text im Audio umwandeln

## 2.2 Beschreibung der Implementierung

Beschreibt 2.2.1 eine SW-Funktion oder ein Arbeitspaket?

### 2.2.1 OpenHABian auf Raspberry PI installieren

Diese Funktion ist sehr wichtig, da sie von allen geplanten Funktionen benötigt wird. OpenHAB wird als zentraler Steuer und auch als Verbinder zwischen Geräten und Benutzern und auch in der Sprachsteuerung benötigt.

Tabelle 4: OpenHABian auf Raspberry PI Tabelle

#	Komponentendetail	Erforderliche Arbeiten
T1	OpenHABian image	Wird verwendet, um openHABian auf Raspberry PI zu installieren
T2	Fernbedienung	Wird verwendet, um Raspberry PI per ssh zu kontrollieren

#### T1: OpenHABian image

- Image wird verwendet, um openHABian auf Raspberry PI zu installieren
- Image muss erstmal auf SD-Karte geflasht werden
- Nach Flashen muss ein File mit Namen "ssh" erstellt werden
- File "ssh" wird für Desktop-Fernbedienung verwendet

#### T2: Desktop-Fernbedienung

- Diese Funktion wird verwendet, um Raspberry PI aus der Ferne zu kontrollieren
- PuTTY wird als Verbindung zwischen Raspberry PI und Laptop per ssh verwendet
- Um PuTTY zu verwenden, wird IP Address von Raspberry PI gefordert

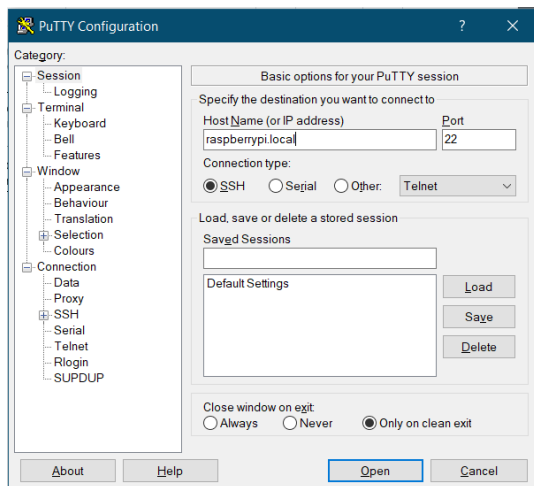


Abbildung 5: PuTTY konfigurieren

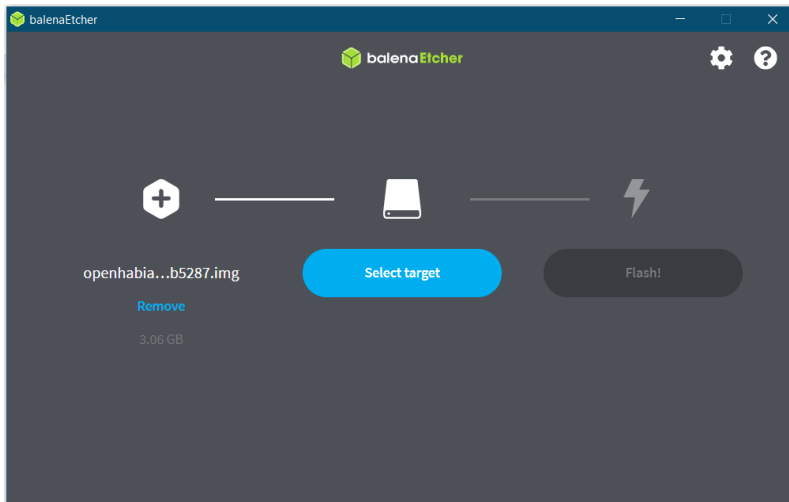


Abbildung 6: Flash via Etcher

## 2.2.2 MQTT

siehe oben

Diese **Funktion** wird für die geplante Funktion benötigt, für die Verbindung zwischen Geräten und openHAB sein.

Tabelle 5: MQTT Tabelle

#	Komponentendetail	Erforderliche Arbeiten
T3	openHAB	wird als zentraler Steuer verwendet
T4	MQTT Bindung	Wird als Verbinder zwischen meisten Geräten und openHAB verwendet

### T3: openHAB

- openHAB ist der zentralen Steuer und auch als Verbinder zwischen Geräten und Benutzer\*in
- openHAB-Konto wurde erstellt (Username="projekt2022", passwort="aldokece")
- openHAB bietet viele Bindungen zur Verbindung mit vielen Marken
- Bei diesem Sprint wird MQTT Bindung verwendet

### T4: MQTT Bindung

- MQTT Bindung wird als Verbinder zwischen openHAB und Geräte verwendet, die ESP8266/ESP8258 Wifi Microchip nutzen
- Vor der Verwendung von MQTT Bindong muss Mosquitto zuerst auf Raspberry installiert werden

## 2.2.3 Geräte per MQTT verbinden

Diese **Funktion** wird für die geplante Funktion benötigt, die eine Verbindung zwischen Geräte und openHAB sein soll.

Tabelle 6: Geräte per MQTT verbinden

#	Komponentendetail	Erforderliche Arbeiten
T5	Smart Geräte	zu steuerndes Gerät
T6	openHAB	wird als zentraler Steuer verwendet
T7	MQTT Bindung	Wird als Verbinder zwischen Geräte und openHAB verwendet

### T5: Smart Geräte

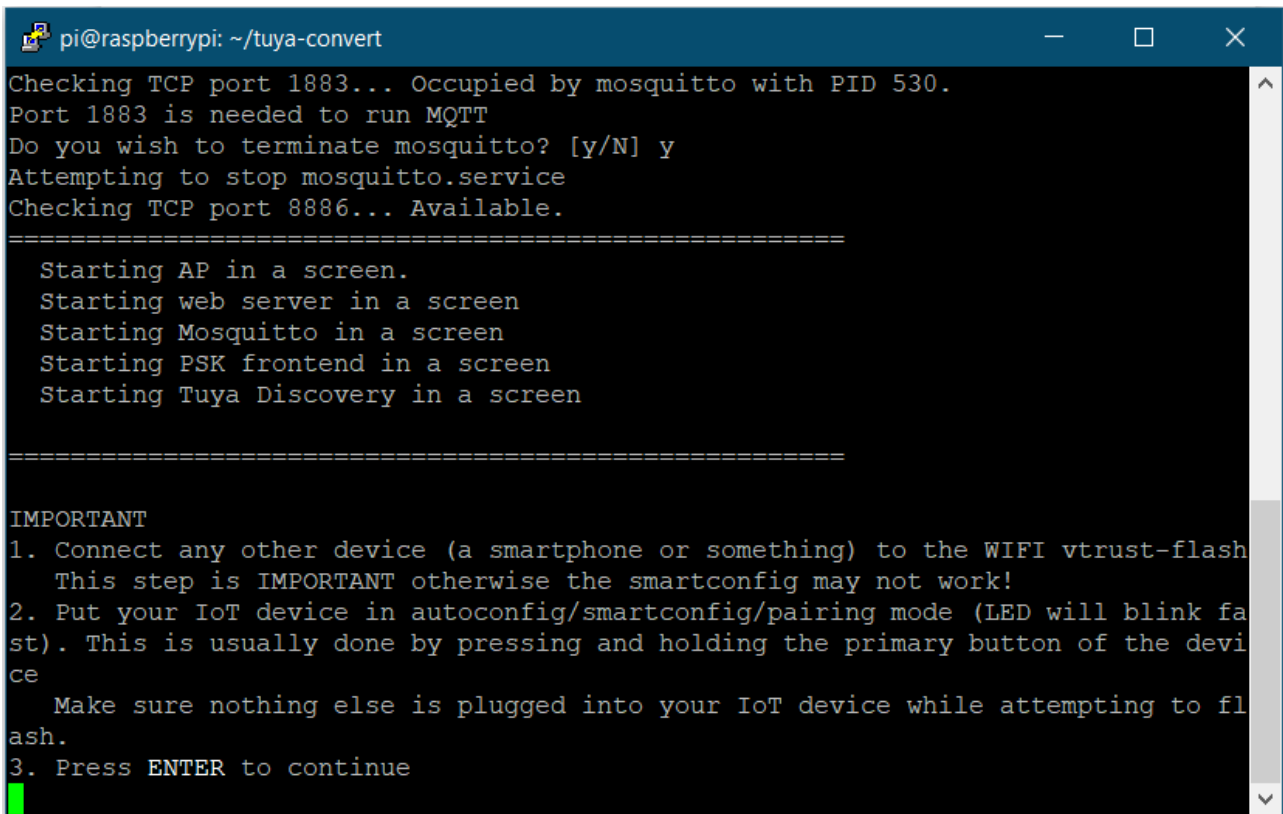
- Smart Geräte sind Geräte mit der Möglichkeit einer Verbindung zum Internet
- Smart Geräte führen Aktionen basierend auf Befehle aus
- Nicht alle Smart Geräte können über MQTT verbunden werden

### T6: openHAB

- openHAB kann verbundene Gerät verwalten
- Dinge, die openHAB je nach Gerätetypen verwalten kann

### T7: MQTT Bindung

- MQTT Bindung wird als Verbinder zwischen openHAB und Smart Geräte verwendet
- Einige Smart Geräte, die ESP8266/ESP8258 Wifi Microchip nutzen, aber nicht mit MQTT kompatibel sind, können mit Tuya-Convert kompatibel gemacht werden



```
pi@raspberrypi: ~/tuya-convert
Checking TCP port 1883... Occupied by mosquitto with PID 530.
Port 1883 is needed to run MQTT
Do you wish to terminate mosquitto? [y/N] y
Attempting to stop mosquitto.service
Checking TCP port 8886... Available.

=====
Starting AP in a screen.
Starting web server in a screen
Starting Mosquitto in a screen
Starting PSK frontend in a screen
Starting Tuya Discovery in a screen

=====

IMPORTANT
1. Connect any other device (a smartphone or something) to the WIFI vtrust-flash
   This step is IMPORTANT otherwise the smartconfig may not work!
2. Put your IoT device in autoconfig/smartconfig/pairing mode (LED will blink fast). This is usually done by pressing and holding the primary button of the device
   Make sure nothing else is plugged into your IoT device while attempting to flash.
3. Press ENTER to continue
```

Abbildung 7: Gerät wird mit Tuya-Convert konfiguriert

## 2.3 System Infrastruktur

Die wichtigste Infrastruktur in diesem Projekt ist openHAB als zentraler Steuer und auch als Verbinder zwischen Geräten und Benutzer\*in. openHAB bietet viele Bindungen zur Verbindung mit vielen gängigen Geräten wie Samsung, Philips HUE, Bluetooth usw. In diesem Sprint haben wir die MQTT Bindung verwendet. In Sprint 2 werden wir weitere erforderliche Bindungen hinzufügen.

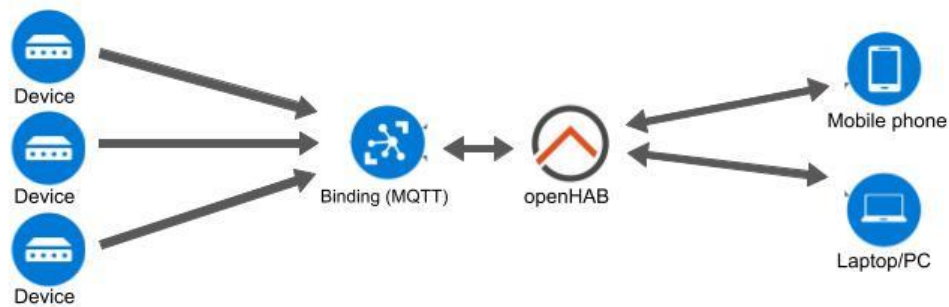


Abbildung 8: System Infrastruktur

### 3 Technische Spezifikation Konstruktion

#### 3.1 Baugruppen

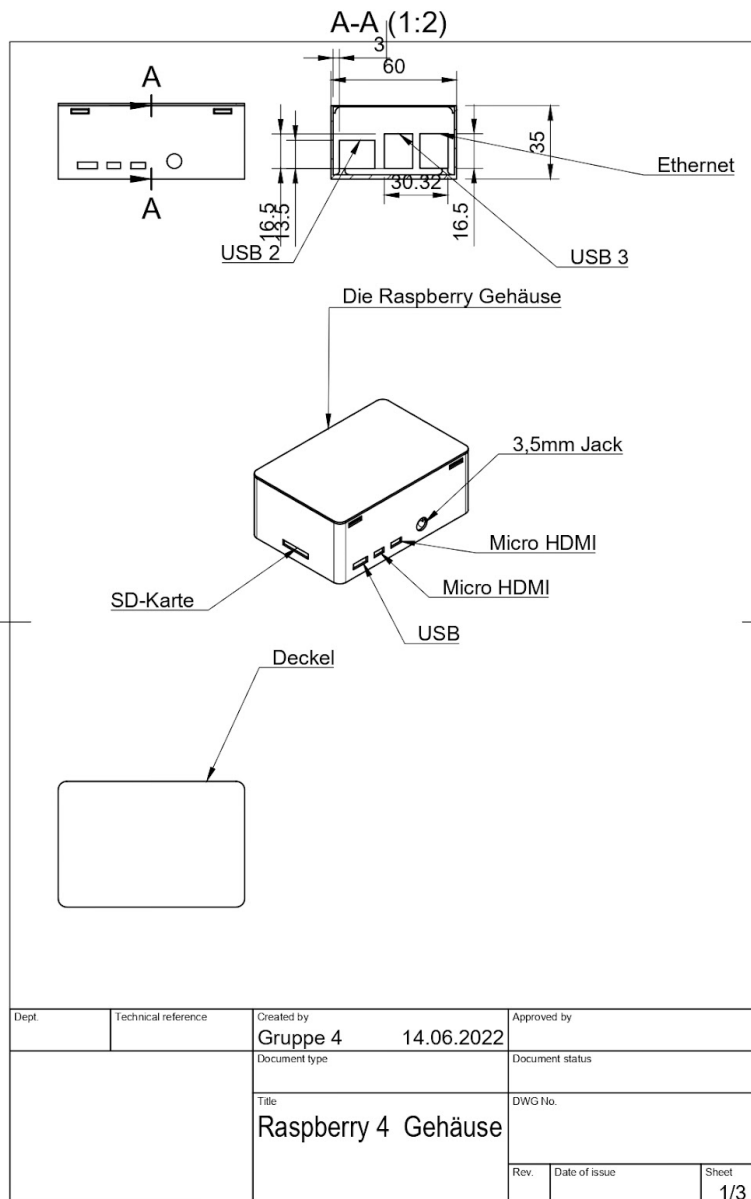


Abbildung 9: Technische Zeichnung

## 4 Modul Abhängigkeiten

Tabelle 7: Modul Abhängigkeiten

#	Name	Abhängig von
1	Raspberry Pi 4	Stromversorgung für die andere Module, wird durch das Ladekabel direkt mit Strom versorgt. Steuert und regelt alle Module.
2	ReSpeaker 4-Mics	Wird mit Raspberry Pi verbunden fürs Signal um Ein- & Ausschalten und auch um die Dateien zu tauschen. Wurde auch von Raspberry Pi 4 das Strom versorgt.
3	Ladekabel	Netzteil versorgt mit Strom
4	LED	Wird mit Raspberry Pi verbunden fürs Signal um Ein- & Ausschalten. Das Raspberry Pi 4 versorgt der Strom.