



# DIPLOMARBEIT

## Aufbau einer Smart-Home-Lösung

Diplomand: Pascal Steiner  
Schule: HF Teko Olten  
Klasse: O-TIN-20  
Ausbildung: Techniker HF Informatik  
Jahr: 2023

## Management-Summary

Die vorliegende Arbeit wurde dem Aufbau einer eigenen Smart-Home-Lösung gewidmet, welche jederzeit in beliebigen Wohnungen eingesetzt werden können soll. Das Thema wurde aufgrund folgender Ausgangslage gewählt:

Erstens sind auf dem aktuellen Markt zahlreiche Hersteller aktiv, welche viele eigene Ecosystems aufgebaut haben. Dies soll nicht unterstützt werden; das Ideal bildet das Vereinen aller Funktionen in einem System.

Zweitens habe ich bereits einige smarte Lichter ausprobiert. Ich wollte ihre Verwendung automatisieren, um Energie zu sparen und meiner Vergesslichkeit entgegenzuwirken: Ich vergesse häufig, das Licht auszuschalten oder ob ich das Fenster geschlossen habe.

Drittens habe ich bei Freunden schon ähnliche Lösungen gesehen. Ich wollte sie auch bei mir einsetzen.

Viertens musste ich eine Lösung erarbeiten, mit der bestehende Hauselektrik erweitert wird, die aber keine deutlichen Umbaumaßnahmen erfordert. Letztere könnten in zukünftigen Wohnungen nicht erlaubt sein.

Für die Umsetzung wurde als Erstes der Markt analysiert. In diesem Rahmen wurden Informationen über die verfügbaren Technologien gesammelt. Danach wurde eine Präferenzmatrix erstellt, um die zu verwendenden Komponenten zu bestimmen. Ferner wurde ein Architekturplan erarbeitet und als Grundlage der Umsetzung gewählt. Bei der Realisierung der Automatisierung dienten die gewünschten Use-Cases als Ausgangspunkt. Sie könnten somit nun weiter ausgearbeitet werden.

Im Rahmen des Projekts wurde Folgendes erkannt: Es sind viele Standards und Geräte auf dem Markt vorhanden, was abschreckend wirken kann. Jedoch ist die Plattform Home Assistant gut als Basis geeignet, denn die meisten Funktionen sind hier einfach integrierbar. Somit wurde ein eigenes System mittels Raspberry Pi und Home Assistant aufgebaut. Zukünftig sollte es noch einfacher werden, da die meisten grossen Player einheitliche Standards zu etablieren beginnen. Der Prototyp konnte über 90 % der Ziele abdecken und kann für die weitere Entwicklung wiederverwendet werden.

Für die weitere Entwicklung müssten mehr Smart Devices ins Netzwerk eingebunden und weitere Automatisierungen vorgenommen werden. Ich werde das System in meiner neuen Wohnung in Betrieb nehmen und gegebenenfalls anpassen oder erweitern. Zudem darf ich das System für meine Eltern aufbauen und implementieren.

# 1 Inhaltsverzeichnis

Management-Summary .....	1
1 Inhaltsverzeichnis.....	2
2 Kurzer beruflicher Lebenslauf .....	4
3 Qualifikationsprofil .....	5
4 Projektinitialisierung .....	7
5 Projektplanung .....	9
5.1 Projektstrukturplanung .....	9
5.2 Projektablaufplanung .....	10
5.3 Risikoanalyse .....	11
6 Projektrealisierung .....	12
6.1 Marktanalyse .....	12
6.2 Technologiebeschrieb.....	14
6.2.1 Marktführer .....	14
6.2.2 Übersicht .....	14
6.2.3 ISO-Standards für Smart-Home-Geräte .....	15
6.2.4 Verwendete Standards.....	16
6.3 Evaluation der geeignetsten Technologie.....	17
6.4 Hauptkriterien.....	17
6.4.1 Präferenzmatrix .....	17
6.4.2 Architekturaufbau.....	19
6.4.3 Verwendete Geräte .....	21
7 Umsetzung des Smart-Home-Projekts .....	25
7.1 Funktionen von Home Assistant .....	25
7.2 Aufbau von Home Assistant .....	29
7.3 Grundkonfiguration von Home Assistant.....	35
7.4 Smart-Geräte-Set-up.....	41
7.5 Smart-Geräte-Integration .....	46
7.6 Integration der Sprachassistenten Google Home und Alexa.....	51
7.6.1 Google Assistant (Nest Mini).....	52
7.6.2 Amazon Alexa.....	54
7.7 Erstellung von Automatisierungen.....	56
7.8 Dashboard konfigurieren.....	58
7.9 Zusammenfassung der Umsetzung.....	62
8 Projektabschluss.....	63
8.1 Projektüberwachung.....	63
8.2 Evaluation der Zielerreichung.....	63
8.2.1 Definition der Testfälle .....	63
8.2.2 Auswertung der Testfälle .....	70
8.3 Reflexion des Wegs zum Ziel .....	71
8.4 Lessons learned .....	72
8.5 Ausblick .....	72

Abkürzungsverzeichnis .....	73
Abbildungsverzeichnis .....	74
Tabellenverzeichnis.....	76
Literatur- und Quellenverzeichnis.....	77
Anhang .....	79
8.6 Anhang 1: Pflichtenheft .....	79
8.7 Anhang 2: Projektstatusberichte.....	86
8.8 Anhang 3: YAML Code Automatisierungen und Konfiguration.....	89
Eigenständigkeitserklärung .....	100

## 2 Kurzer beruflicher Lebenslauf

# PASCAL STEINER

INFORMATIKER

pascal-steiner@lorenz-steiner.ch  
Belchenstrasse 7  
+41 763437705  
14 Nov, 1997



### BETREFF

"Ein hochqualifizierter Informatiker mit umfassender Erfahrung in der Entwicklung und Implementierung von IT-Lösungen. Expertise in Programmiersprachen, Datenbanken und Netzwerken. Zielorientiert und teamfähig."

### KOMPETENZEN

Netzwerke

Agiles Projektmanagement

Linux

Virtualisierungen

Docker

Infrastructure as Code

Monitoring - Zabbix

Windows Server & Clients

Office365 / Azure / Intune

### SPRACHKENNTNISSE

Deutsch | Muttersprache

English | Fortgeschritten.

Cambridge Advanced Certificate

### BERUFSERFAHRUNG

**System Engineer** Apr 2023 – Aktuell  
UMB AG – Cham

- Erfahrung in der Konzeption und Implementierung von IT-Infrastrukturen und Netzwerklösungen.
- Erfahrung in der Zusammenarbeit mit Kunden und Projektteams zur Umsetzung von IT-Projekten.
- Erfahrung in der Fehlerbehebung und Optimierung von Systemen zur Steigerung der Leistung.

**System Administrator / Engineer** Nov 2019 – Sep 2022  
Intersim AG – Burgdorf

- Betrieb, Verwaltung und Instandhaltung Client und Server Infrastrukturen
- Verwaltung von Hardware, Lizenzen und Zertifikaten
- First, Second & Third Level Support
- Evaluierungen Partner und IT Komponenten
- Aufbau neuer Cloud Infrastrukturen
- Konzeptionierungen, Dokumentierung und Umsetzungen verschiedenster Projekte
- Betrieb, Wartung, Instandhaltung und Automatisierung der IT Systemen

**ICT System Specialist** Nov 2017 – Jun 2019  
Bundesamt für Informatik und Telekommunikation – Bern

- Manage Linux Webservices der CMS Umgebungen
- Instandhaltung & Migrationen der CMS Lösung
- Second Level Support

**Ausbildung Applikationsentwickler** Aug 2013 – Aug 2017  
terreActive AG – Aarau

- Grundausbildung
- Entwicklung und Wartung von Applikationen
- Betreiben von Webapplikationen und CMS Umgebungen

### AUSBILDUNG

**Höhere Fachschule zum dipl. Techniker Systemtechnik** 2020 – Aktuell  
Teko – Olten

**Ausbildung Informatiker EFZ - Applikationsentwicklung** 2013 – 2017  
BBB - Berufsbildung Baden – Baden

**MCSA Windows Server 2026** 2019 – 2019  
Migros Klubschule – Bern

Abbildung 1 Lebenslauf

### 3 Qualifikationsprofil

<b>Menschen führen</b> Prozess 1	Im Rahmen des neu erstellten Teams durfte ich viele meiner Teamkollegen einarbeiten und die Infrastruktur des Kunden bekanntmachen. Zudem durfte ich viele Freelancer an einer Bar managen, betreuen und unterstützen.
<b>Entscheidungen fällen</b> Prozess 2	Im Rahmen von Migrationsprojekten und Transitionsphasen in der aktuellen Arbeitsumgebung sowie beim letzten Arbeitsgeber musste ich viele Entscheidungen mit weitreichenden Konsequenzen treffen. Ich gelangte unter Einsatz von Präferenzmatrizen, Analysen und Teamfeedback zu ihnen.
<b>Projekte planen und leiten</b> Prozess 3	Mehrere kleine Migrationsprojekte und ein grosses Vorhaben im Bereich Webhosting, Serverinfrastruktur sowie Inhouse-Netzwerk habe ich allein sowie in Zusammenarbeit mit Teamkollegen geplant und erfolgreich umgesetzt.
<b>Sich sprachlich verständigen</b> Prozess 4	Ich habe mehrere Projekt- und Systemdokumentationen sowie adressatengerecht dem Team und den Projektleitern kommuniziert.
<b>Unternehmensprozesse verstehen und mitgestalten</b> Prozess 6	Im Rahmen der <holokratischen Hierarchie> der Intersim AG durfte ich viele Rollen und Kreise überarbeiten, erweitern sowie optimieren.
<b>Geschäftsziele erreichen</b> Prozess 7	Durch den aktiven Betrieb und das Monitoring konnten ich und mein Team dem Unternehmen helfen, sein Budget einzuhalten und die Kundenziele zu erreichen. Da wir bestrebt waren, möglichst viel zu automatisieren und zu verbessern, konnten viel Zeit und Kosten eingespart werden.
<b>Probleme analysieren und lösen</b> Prozess 9	Wir haben zahlreiche Hosting- und Systemfehler erkannt sowie analysiert. Im Team haben wir Workarounds und permanente Lösungen erstellt.
<b>Sich persönlich weiterentwickeln</b> Prozess 10	Wir haben regelmässig an Security-Events und -Audits teilgenommen, um uns aktueller Risiken bewusst zu sein. Zudem habe ich eine persönliche Weiterbildung im Bereich Systemtechnik an der HF Teko Olten zum Diplomierten Techniker absolviert.

---

<b>Datenschutz und Datensicherheit gewährleisten</b> Prozess 14	Als Systemadministrator war ich für die interne IT sowie die Kundensysteme verantwortlich. Deswegen haben wir MFA und weitere Sicherheitssysteme eingeführt.
<b>System- und Netzwerkarchitektur bestimmen</b> Prozess 17	Beim aktuellen Arbeitsgeber mussten wir viele Systeme und Netzwerke von der V-ZUG AG erlernen, um den Betrieb zu gewährleisten. Deren Infrastruktur ist kompliziert, weswegen alle Teammitglieder die Infrastruktur verstehen mussten, um die Zusammenhänge zwischen Systemen und Netzen nachzuvollziehen. Aktuell wird das Projekt der Upgrades von Netzwerkkomponenten umgesetzt, da viele Switches veraltet waren.
<b>Konzepte und Dienste entwickeln</b> Prozess 18	Im Rahmen der Migration eines grossen Kunden beim letzten Arbeitgeber durften ich und mein Teamkollege erstmalig mehrere Konzepte mit verschiedenen Hosting-Anbietern sowie Technologien erarbeiten. Danach mussten wir die Planung der Migration erstellen und ein PoC (Proof of Concept) durchführen. Dazu haben wir neue Technologien erlernt und Business-«best-Practises» verwendet.
<b>Konzepte und Dienste umsetzen</b> Prozess 19	Das ausgearbeitete Konzept der Migration durften ich und mein Kollege umsetzen. Wir haben hierbei neue Technologien verwendet und der organisatorische Aufwand war gross. Es wurden Termine und Meilensteine geplant und eingehalten. Nach jedem Meilenstein nahmen wir eine Reflexion vor, um die Prozesse zu verbessern und die Kundenzufriedenheit sicherzustellen. Über einen regelmässigen Austausch mit dem Kunden und den anderen Projektmitgliedern haben wir die Zufriedenheit sowie die Effizienz gesteigert.
<b>Spezifische Hardware programmieren</b> Prozess 20	Als Projektarbeit im Fach Microcomputertechnik habe ich mittels Arduino eine klein angelegte Innenraumüberwachung und Lichtsteuerung erarbeitet. Integriert waren eine Sensorüberwachung und eine Lichtsteuerung mittels Schalter sowie Routinen. Im Rahmen dieser Diplomarbeit habe ich zahlreiche Einbau-sensoren implementiert und mittels Home Assistant sowie Automatisierungen gesteuert.

## 4 Projektinitialisierung

Im Rahmen der Vorbereitungsarbeiten wurde ein Pflichtenheft gemäss Vorgaben erarbeitet. Die vollständige Version ist im Anhang 1 beigelegt. Nachfolgend werden die Ziele zusammengefasst.

### **Genaue Zieldefinition**

Als Ziel galt es, ein Smart-Home-System mit verschiedenen Komponenten und Technologien zu entwickeln.

- Erfolgskriterien:
  - Prototyp eines Smart-Home-Systems mit folgenden Komponenten und Technologien:
    - Zigbee-fähige Geräte
    - Beleuchtungssteuerung
    - Temperaturüberwachung
    - Gerätesteuerung
  - Realisierung von Schlüsselfunktionen:
    - Fernsteuerung der Beleuchtung
    - Überwachung der Raumtemperatur
    - Einbindung von Sicherheitskomponenten
    - Bedienung von Geräten
  - Integration von mindestens drei verschiedenen Zigbee-fähigen Geräten verschiedener Hersteller
  - Sprachsteuerung über Google Assistant und Alexa
  - Entwicklung einer benutzerfreundlichen Oberfläche
  - Durchführung umfassender Tests
  - Dokumentation der Ergebnisse

### **Zusammenführung**

Im Rahmen der Diplomarbeit sollte ein Smart-Home-System mit verschiedenen Komponenten und Technologien entwickelt werden, das Privatanwender einfach und kostengünstig nutzen können.

Das System sollte folgende Funktionen sowie Eigenschaften bieten:

- Automatische Lichtsteuerung: Das Licht sollte automatisch ein- und ausgeschaltet werden, je nach Anwesenheit und Bewegung im Raum.
- Zusätzliche smarte Schalter: Es sollten zusätzliche smarte Schalter für weitere Geräte und Funktionen verfügbar sein.
- Sprachgesteuerte Schaltungen: Geräte und Funktionen sollten mit Sprachbefehlen gesteuert werden können.
- Datensammlung: Daten über den Stromverbrauch, die Temperatur und die Bewegungen im Haus sollten gesammelt werden.
- Sicherheit: Das Haus sollte vor Eindringlingen und anderen Gefahren geschützt werden.
- Spielerische Automatismen: Spielerische Automatismen sollten verfügbar sein, die das Leben im Haus angenehmer machen.
- Grafische Übersicht: Eine grafische Übersicht aller Schalter, Sensoren, Automatismen und Datensammlungen sollte verfügbar sein.

Um diese Ziele zu erreichen, wurden folgende Schritte durchgeführt:

1. Marktanalyse: Es wurde eine Marktanalyse über DIY(Do-it-Yourself)-smart-Home-Geräte durchgeführt.
2. Sensoren und Aktoren: Es wurden verschiedene Sensor- und Aktortypen untersucht. Für diese wurde eine Vergleichsmatrix mit Vor- und Nachteilen erstellt.
3. Aufbau des Smart-Home-Hub und der verknüpften Gateways: Ein Home-Server auf einem Raspberry Pi (z. B. Home Assistant) wurde konfiguriert und ein Zigbee-Gateway sowie Hue-Geräte wurden eingebunden. Google Assistant, Amazon Alexa sowie andere Standalone-WLAN (Wireless Local Area Network)-Devices wurden konfiguriert.
4. Automatisierung/Abläufe: Skripte zur Präsenzerkennung und zur automatischen Geräteabschaltung wurden entwickelt. Routinen bei Verlassen und Betreten von Räumen oder der Wohnung wurden erstellt. Sprachgesteuerte Routinen für Events wurden implementiert.
5. Grafische Oberfläche: Bestehende Tools wurden analysiert und es wurde eine Benutzeroberfläche festgelegt. Die grafische Oberfläche wurde konfiguriert.
6. Globale Erreichbarkeit: Der Zugriff auf die grafische Oberfläche von überall mittels Smartphone-Browser oder einer App wurde ermöglicht. Die Option zur Steuerung von Licht und Steckdosen von überall auf der Welt wurde implementiert.

## 5 Projektplanung

Nachfolgend wird die gesamte Planung des Projektes aufgezeigt. Dazu gehören Struktur- und Ablauf- Plan sowie eine Risikoanalyse.

### 5.1 Projektstrukturplanung

Im Rahmen der Projektstrukturplanung (PSP) wird das Projekt in Arbeitspakete aufgeteilt, welche in Summe die Endergebnisse des Projektes ergeben. Abbildung 2 enthält einen Projektstrukturplan, dargestellt als Strukturbaum.

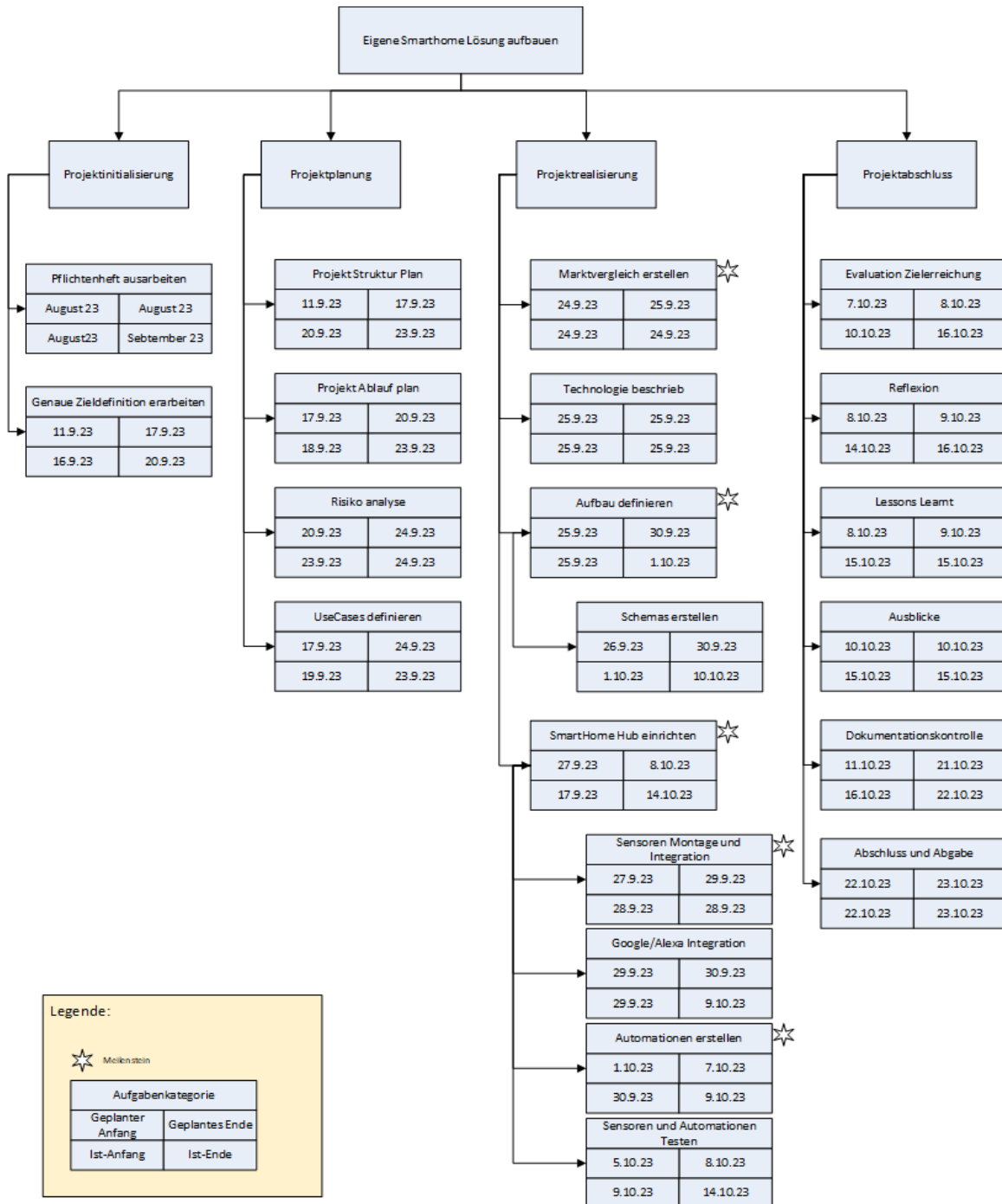


Abbildung 2 Projektstrukturplan



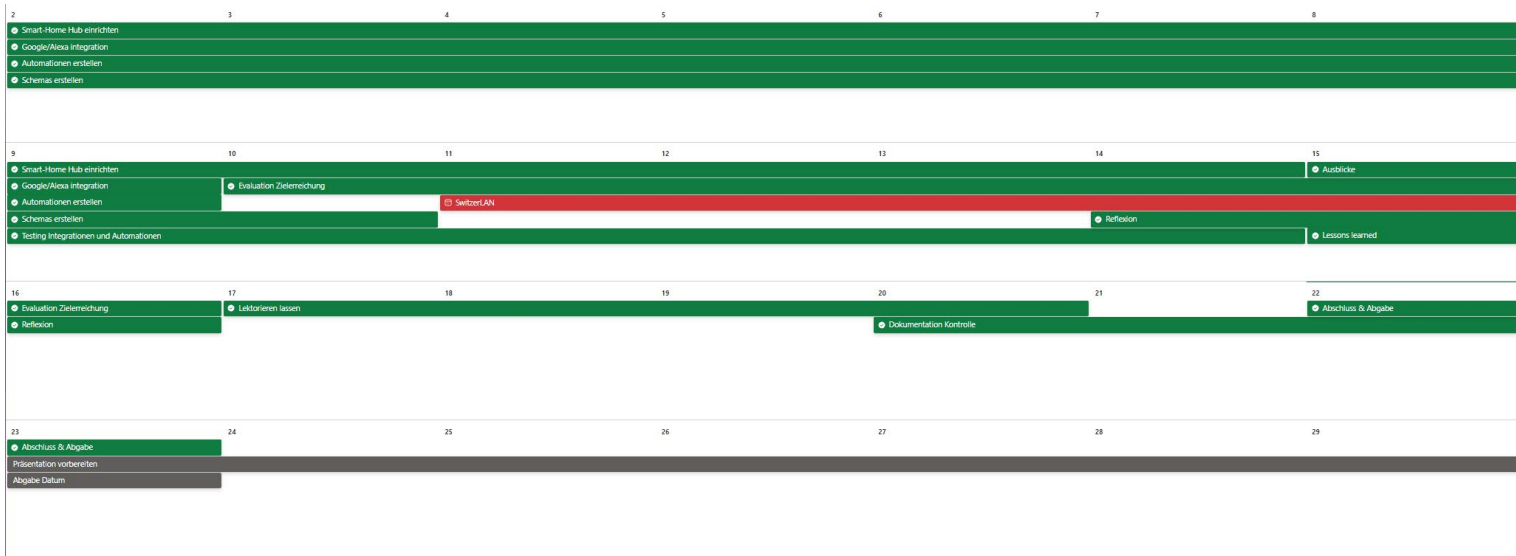


Abbildung 5 IST Planung - Oktober

### 5.3 Risikoanalyse

Die Risiken sind in Kategorien wie technische, zeitliche und finanzielle Risiken unterteilt. Ihre Eintrittswahrscheinlichkeit und ihre Auswirkungen sind von niedrig bis hoch kategorisiert.

Risiko	Kategorie	Wahrscheinlichkeit	Auswirkung	Massnahmen zur Risikominimierung
Technisches Scheitern	technisch	mittel	hoch	kontinuierliche Überwachung und Fehlerbehebung während der Entwicklung
Zeitverzögerung	zeitlich	mittel-hoch	hoch	Erstellung eines detaillierten Zeitplans und regelmässige Fortschrittsüberwachung
Budgetüberschreitung	finanziell	niedrig	mittel	Genauigkeit der Kostenprognosen überprüfen und zusätzliche Finanzmittel bereithalten
Datenverlust	technisch	niedrig	mittel	Implementierung von regelmässigen Datensicherungen und Wiederherstellungsverfahren

Tabelle 1 Risikoanalyse

## 6 Projektrealisierung

In diesem Kapitel geht es um die Umsetzung des Projekts.

Aufgeteilt mit dem Theoretischen und Informativen Teil zu Beginn und danach mit der praktischen Umsetzung.

### 6.1 Marktanalyse

Der Markt für Smart-Home-Hubs ist in den letzten Jahren stark gewachsen. In einer Studie von Statista wird der 2025 weltweit mit Smart-Home-Hubs erzielte Umsatz auf 12,5 Milliarden US-Dollar geschätzt. Die Treiber dieses Wachstums sind die zunehmende Popularität von Smart-Home-Geräten, die steigende Nachfrage von intelligenten Funktionen und die Verfügbarkeit von erschwinglichen Hubs. In der nachfolgenden Marktanalyse werden die wesentlichen Marktteilnehmer, Trends und Herausforderungen von Smart-Home-Hubs untersucht.

#### Marktübersicht

Der Markt für Smart-Home-Hubs ist in zwei Hauptsegmente unterteilt, die nachfolgend vorgestellt werden.

**Closed-Source-Hubs:** Diese Hubs werden von einem einzigen Unternehmen entwickelt und vermarktet. Sie sind in der Regel mit einer begrenzten Auswahl an Smart-Home-Geräten kompatibel.

**Open-Source-Hubs:** Diese Hubs werden von der Community entwickelt und können mit einer Vielzahl von Smart-Home-Geräten kompatibel sein.

Die Marktführer im Bereich Smart-Home-Hubs sind Amazon, Google, Apple und Samsung. Diese Unternehmen bieten eine breite Palette von Hubs an, die mit einer Vielzahl von Smart-Home-Geräten kompatibel sind.

#### Trends

Folgende Trends sind im Markt für Smart-Home-Hubs zu beobachten:

**Erweiterung der Plattformunterstützung:** Die meisten Hubs sind mittlerweile mit einer Vielzahl von Smart-Home-Geräten kompatibel. Dies vereinfacht es Nutzern, ein Smart Home aufzubauen, das aus Geräten verschiedener Hersteller besteht.

**Integration von Sprachassistenten:** Sprachassistenten wie Amazon Alexa und Google Assistant werden immer beliebter. Viele Hubs ermöglichen ihre Integration, sodass Nutzer ihre Smart-Home-Geräte mit Sprachbefehlen steuern können.

**Automatisierung:** Smart-Home-Hubs können zur Automatisierung von Aufgaben verwendet werden. So können Nutzer beispielsweise ihre Beleuchtung einschalten, wenn sie nach Hause kommen, oder ihre Heizung ausschalten, wenn sie das Haus verlassen.

#### Herausforderungen

Die folgenden Herausforderungen sind im Markt für Smart-Home-Hubs zu beobachten:

**Sicherheit:** Smart-Home-Hubs können ein potenzielles Sicherheitsrisiko darstellen. Nutzer sollten sich daher über die Sicherheitsfunktionen ihres Hubs informieren.

**Kompatibilität:** Die Kompatibilität zwischen Smart-Home-Hubs und -Geräten kann ein Problem darstellen. Nutzer sollten daher vor dem Kauf prüfen, ob ihr Hub mit den Geräten kompatibel ist, die sie verwenden möchten.

**Preis:** Smart-Home-Hubs können teuer sein. Nutzer sollten dementsprechend vor dem Kauf einen Vergleich der Preise durchführen.

**Ausblick**

Der Markt für Smart-Home-Hubs wird in den nächsten Jahren weiterwachsen. Die zunehmende Popularität von Smart-Home-Geräten und die Entwicklung neuer Technologien dürften zu diesem Wachstum beitragen.

**Fazit**

Der Markt für Smart-Home-Hubs ist dynamisch und entwickelt sich kontinuierlich weiter. Nutzer sollten sich vor dem Kauf eines Hubs über die verschiedenen Marktteilnehmer, Trends und Herausforderungen informieren.

In Tabelle 2 folgt ein Vergleich verschiedener Hubs.

Produkt	Unterstützte Technologien	Integrationen/Funktionen	Preis
Google Home	WiFi, Bluetooth	Sprachsteuerung, Zeitpläne und Automatisierungen, Musik- und Video-Streaming	ab 59,99 €
Amazon Alexa	WiFi, Bluetooth	Sprachsteuerung, Zeitpläne und Automatisierungen, Musik- und Video-Streaming	ab 39,99 €
Samsung Smart-Things	WiFi, Bluetooth, Zigbee, Z-Wave	Sprachsteuerung, Zeitpläne und Automatisierungen, Musik- und Video-Streaming	ab 49,99 €
Apple HomeKit	WiFi, Bluetooth	Sprachsteuerung, Zeitpläne und Automatisierungen, Musik- und Video-Streaming	ab 99,99 €
Home Assistant	WiFi, Bluetooth, Zigbee, Z-Wave	Sprachsteuerung, Zeitpläne und Automatisierungen, Musik- und Video-Streaming, Heizungs-, Klima- und Zutrittssteuerungen	kostenlos
OpenHub	WiFi, Bluetooth, Zigbee, Z-Wave	Sprachsteuerung, Zeitpläne und Automatisierungen, Musik- und Video-Streaming, Heizungs- und Klimasteuerungen, Zutrittssteuerungen	kostenlos
Philipps Hue	WiFi, Zigbee	Zeitpläne und Automatisierungen	ab 39,90 €

*Tabelle 2 Smart-Home-Hubs im Vergleich*

WiFi steht für Wireless Fidelity. Dies ist eine Bezeichnung für Geräte, die in der Lage sind WLAN zu empfangen.

## 6.2 Technologiebeschreibung

### 6.2.1 Marktführer

Momentan verfügbar sind auf dem Markt primär Z-Wave-, Zigbee- und WiFi-Technologien. Es gibt auch Hersteller, die andere Interpretationen und Integrationen der in Tabelle 2 genannten Technologien nutzen, aber sie sind wenig verbreitet.

Im Oktober 2022 wurde Matter vorgestellt. Dabei handelt es sich um einen allgemeinen Standard, an dem sich alle Hersteller orientieren können, um Kompatibilität und Interkonnektivität zu gewährleisten.

### 6.2.2 Übersicht

Nachfolgend werden nun die meistverbreiteten Technologien aufgeführt und vorgestellt.

#### Zigbee

- Frequenz: 2,4 GHz
- Reichweite: bis zu ca. 100 Meter in Sichtlinie, 30 Meter in Innenräumen
- Datenrate: 250 kbit/s
- Mesh-Netzwerk: ja
- Kompatibilität: offen, Zigbee Alliance
- Vorteile:
  - besonders stromsparend
  - hohe Datensicherheit
  - Mesh-Netzwerk für eine zuverlässige Verbindung
  - geeignet für grosse Smart-Home-Netzwerke
- Nachteile:
  - kann durch andere 2,4-GHz-Geräte gestört werden
  - keine volle Kompatibilität zwischen allen Geräten
  - relativ niedrige Datenrate
  - unvollständige Kompatibilität zwischen Zigbee-Geräten

#### Z-Wave

- Frequenz: 908 MHz
- Reichweite: bis zu ca. 30 Meter in Sichtlinie, 10 Meter in Innenräumen
- Datenrate: 9,6 kbit/s
- Mesh-Netzwerk: ja
- Kompatibilität: geschlossen, Z-Wave Alliance
- Vorteile:
  - besonders zuverlässig und stabil
  - geeignet für grosse Smart-Home-Netzwerke
  - keine Störung durch andere Geräte
- Nachteile:
  - weniger stromsparend als Zigbee
  - höhere Kosten für Geräte als bei Zigbee
  - geringere Reichweite als bei Zigbee.
  - relativ niedrige Datenrate
  - ein in der Regel höherer Kostenpunkt für Z-Wave-Geräte als für Zigbee-Geräte

## WLAN

- Frequenz: 2,4 GHz oder 5 GHz
- Reichweite: bis zu 100 Meter in Sichtlinie, 30 Meter in Innenräumen
- Datenrate: bis zu 1 GBit/s
- Mesh-Netzwerk: nein
- Kompatibilität: offen
- Vorteile:
  - hohe Datenrate
  - geeignet für alle Geräte, die eine Internetverbindung erfordern
- Nachteile:
  - weniger stromsparend als Zigbee oder Z-Wave
  - mögliche Störung durch andere WLAN-Geräte
  - Notwendigkeit vieler IP-Adressen(Adresse in Computernetzwerken die auf das Internetprotokoll basieren) bei vielen Geräten (kann herkömmliche DHCP(Dynamic Host Configuration Protocol)-Konfigurationen übermässig beanspruchen)

## Matter

- Frequenz: 2,4 GHz oder 5 GHz
- Reichweite: bis zu 100 Meter in Sichtlinie, 30 Meter in Innenräumen
- Datenrate: bis zu 1 GBit/s
- Mesh-Netzwerk: ja
- Kompatibilität: offen
- Vorteile:
  - volle Kompatibilität zwischen allen Matter-Geräten
  - hohe Datensicherheit
  - geeignet für alle Smart-Home-Geräte
- Nachteile:
  - noch nicht weit verbreitet

## Geräte/Protokolle

- Zigbee: Beleuchtung, Sensoren, Rauchmelder, Türschlösser, Rollläden
- Z-Wave: Beleuchtung, Sensoren, Rauchmelder, Türschlösser, Rollläden, Heizungssteuerung
- WLAN: alle Geräte, die eine Internetverbindung aufweisen
- Matter: alle Smart-Home-Geräte

### 6.2.3 ISO-Standards für Smart-Home-Geräte

Die International Organization for Standardization (ISO) hat eine Sammlung an Standards für Smart-Home-Geräte veröffentlicht. Diese Standards sollen die Interoperabilität zwischen Geräten unterschiedlicher Hersteller gewährleisten und die Sicherheit sowie die Zuverlässigkeit von Smart-Home-Systemen steigern.

#### ISO 33000-3

Dieser Standard bezieht sich auf Anforderungen an die Sicherheit von Smart-Home-Geräten. Er umfasst Aspekte wie die Authentifizierung von Geräten, die Verschlüsselung von Daten und Schutzmassnahmen gegen Missbrauch.

**ISO 33000-4**

Dieser Standard enthält Anforderungen an die Interoperabilität von Smart-Home-Geräten. Konkretisiert werden Aspekte wie Datenformate, Protokolle und Schnittstellen für die Kommunikation zwischen Geräten.

**ISO 33000-5**

In diesem Standard werden Anforderungen an die Zuverlässigkeit von Smart-Home-Geräten definiert. Er umfasst Aspekte wie die Lebensdauer von Geräten, die Wartungsfreundlichkeit und die Widerstandsfähigkeit gegen Störungen.

**Weitere Standards**

Neben den ISO-Standards gibt es auch eine Reihe von anderen Standards für Smart-Home-Geräte, beispielsweise die folgenden:

- IEEE 802.15.4
- Zigbee
- Z-Wave
- Thread

**6.2.4 Verwendete Standards**

Zigbee unterstützt die folgenden ISO-Standards:

- ISO 33000-3: Sicherheit
- ISO 33000-4: Interoperabilität
- ISO 33000-5: Zuverlässigkeit

Z-Wave unterstützt die folgenden ISO-Standards:

- ISO 33000-3: Sicherheit
- ISO 33000-4: Interoperabilität
- ISO 33000-5: Zuverlässigkeit

Matter ist ein neuer Standard, der die Interoperabilität zwischen Smart-Home-Geräten unterschiedlicher Hersteller gewährleisten soll. Matter unterstützt die folgenden ISO-Standards:

- ISO 33000-3: Sicherheit
- ISO 33000-4: Interoperabilität
- ISO 33000-5: Zuverlässigkeit

Die WLAN-Technologie basiert auf dem IEEE-802.11-Standard, der vom IEEE (Institute of Electrical and Electronics Engineers) entwickelt wurde. Im Standard werden die physikalische Schicht (PHY) und die Medium-Access-Control (MAC) für WLAN-Netzwerke definiert.

Ferner wird WLAN von der ISO (International Organization for Standardization) als ISO/IEC 8802-11 standardisiert. Die ISO-/IEC-8802-11-Norm ist identisch mit dem IEEE-802.11-Standard.

Zusätzliche Standards:

Neben den ISO-Standards unterstützen Zigbee und Z-Wave auch eine Reihe von anderen Standards, beispielsweise die folgenden:

- IEEE 802.15.4: PHY- und MAC-Layer
- IEEE 802.15.4e: Mesh-Networking
- IEEE 802.15.4g: Power-Saving

Matter basiert auf dem IEEE-802.15.4-Standard und unterstützt Mesh-Networking.

### 6.3 Evaluation der geeignetsten Technologie

In Nachfolgenden unter Kapiteln wird aufgezeigt wie zur Präferierten Technologie gefunden wurde. Zudem wird der Architekturaufbau dargestellt und die verwendeten Geräte aufgelistet.

### 6.4 Hauptkriterien

Welche Funkstandards die beste Wahl darstellen, hängt von den individuellen Anforderungen ab. Zigbee und Z-Wave sind jeweils zuverlässig und stabil, wobei Zigbee stromsparender ist und Z-Wave eine grössere Reichweite hat. Die beste Wahl für Geräte, die eine Internetverbindung benötigen, bildet WLAN, jedoch sind Zigbee sowie Z-Wave stromsparender. Matter ist der neueste Standard und bietet eine volle Kompatibilität zwischen allen Geräten.

Für den Einsatz in einem grossen Smart-Home-Netzwerk mit vielen Geräten ist Zigbee oder Z-Wave die beste Wahl. Für Geräte, die eine Internetverbindung aufweisen, sollte WLAN genutzt werden. Matter eignet sich für neue Geräte, die mit einem bestehenden Smart-Home-Netzwerk kompatibel sein sollen.

Zudem sollten die Kosten und die Verfügbarkeit im Einzelfall geprüft werden sowie die Entscheidung mitbestimmen.

Dies kann bei jedem anders gewertet werden. Im Rahmen dieser Arbeit werden bereits vorhandene WLAN-Geräte integriert und danach möglichst auf einen Standard ausgerichtet. Für mich bildeten die Kosten den zentralen Entscheidungsfaktor, weshalb primär Zigbee-Geräte eingekauft wurden. Dies geschah im Vorfeld des Projekts, weil ich viele Geräte über Aliexpress direkt aus China bezogen habe und dabei lange Lieferzeiten auftreten können.

Im nachfolgenden Abschnitt befindet sich eine Präferenzmatrix, aus der die Entscheidung für oder gegen eine Technologie abgeleitet werden kann.

#### 6.4.1 Präferenzmatrix

In der Präferenzmatrix in Tabelle 3 werden die Technologien Zigbee, Z-Wave, WLAN und Matter anhand ihrer Kosten sowie ihrer Verfügbarkeit bewertet. Die Einordnung erfolgt auf einer Skala von eins (niedrig) bis fünf (hoch).

Technologie	Kosten (1–5)	Verfügbarkeit (1–5)	Gesamtbewertung (1–10)
Zigbee	5	4	9
Z-Wave	3	3	6
WLAN	2	5	7
Matter	4	2	6

Tabelle 3 Präferenzmatrix

Erläuterungen zur Bewertung:

- **Kosten (1–5):**
  - Zigbee erhält eine hohe Bewertung von fünf, da die Anwendung kostengünstig ist, insbesondere wenn Geräte aus China bezogen werden.
  - Z-Wave wird mit drei bewertet, da die Geräte in der Regel teurer sind als Zigbee-Alternativen.

- WLAN wird mit zwei bewertet, zumal WLAN-Geräte in der Regel teurer sind als die Alternativen und bei vielen Geräten zusätzliche IP-Adressen erforderlich sein können.
- Matter wird mit vier bewertet, da der Standard noch nicht weit verbreitet ist und die Preise variieren können.
- Verfügbarkeit (1–5):
  - Zigbee erhält eine Bewertung von vier, da die Verfügbarkeit von Geräten verschiedener Hersteller gut ausfällt, aber es standortbezogene Unterschiede in der Verfügbarkeit geben kann.
  - Z-Wave wird mit drei bewertet, da die Verfügbarkeit von Geräten begrenzter sein kann als bei den Alternativen, zumal es sich um einen geschlossenen Standard handelt.
  - WLAN erhält eine hohe Bewertung von fünf, weil WLAN-Geräte weit verbreitet und leicht verfügbar sind.
  - Matter wird mit zwei bewertet, da diesem Standard entsprechende Lösungen aufgrund seiner Neuartigkeit möglicherweise noch nicht in grosser Vielfalt erhältlich sind.
- Gesamtbewertung (1–10):
  - Die Gesamtbewertung ergibt sich aus der Summe der Bewertungen von Kosten und Verfügbarkeit. Zigbee erhält die höchste Gesamtbewertung von neun, was die Option zur bevorzugten Technologie für das Projekt macht. Für WLAN ergibt die Gesamtbewertung sieben, gefolgt von Z-Wave und Matter mit einer Gesamtbewertung von sechs.

Diese Präferenzmatrix half bei der Entscheidung für eine Technologie, wobei Zigbee aufgrund der Kombination aus niedrigen Kosten und guter Verfügbarkeit als bevorzugte Option hervorgeht. Dennoch wurden vorliegend auch bestehende Geräte integriert; sie wurden nicht ersetzt.

### 6.4.2 Architekturaufbau

Nachfolgend ist eine erste grobe Übersicht des geplanten Aufbaus enthalten.

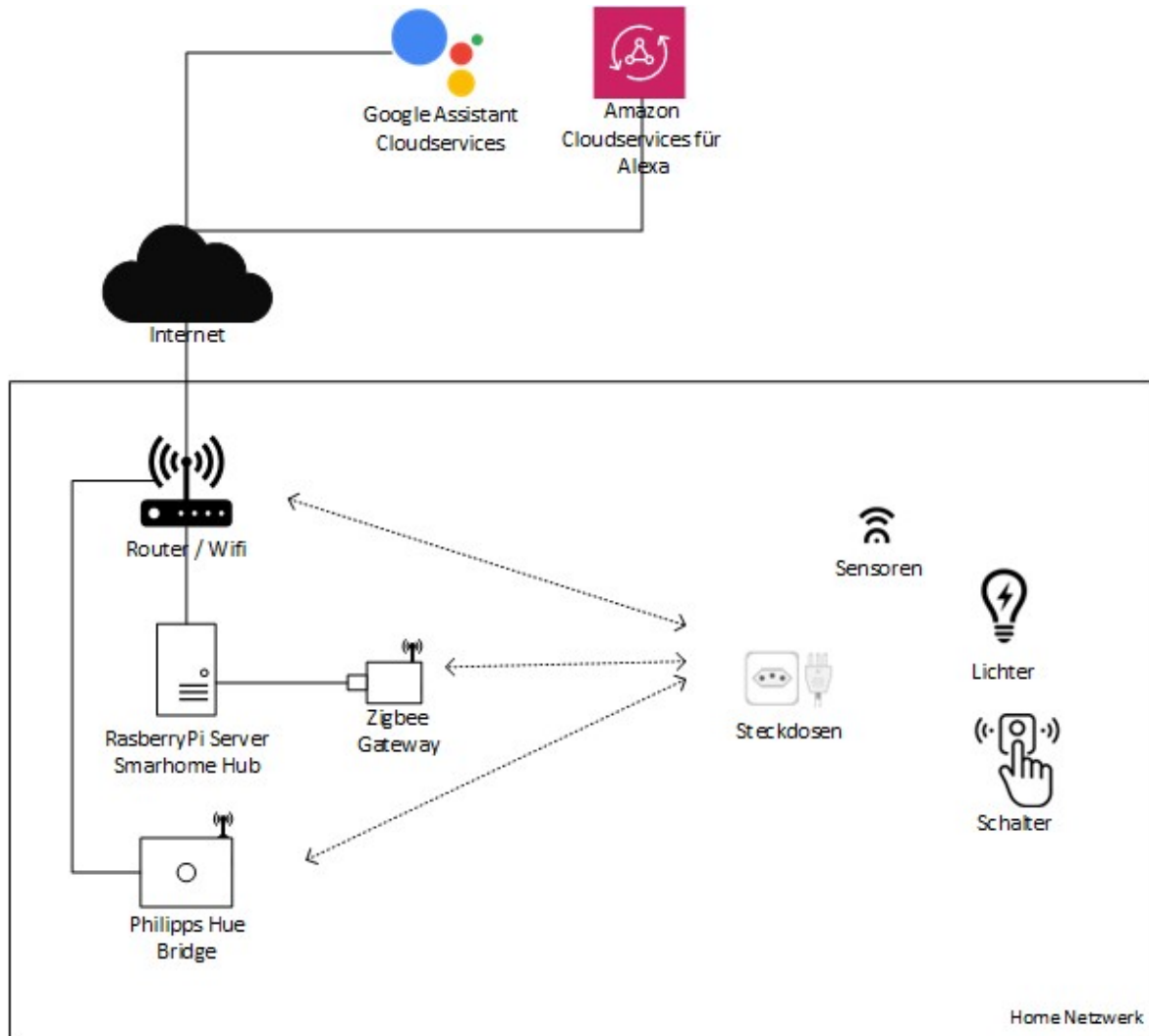


Abbildung 6 Basis-Architekturplan

Durch die Umsetzung ist der nachfolgende detaillierte Plan entstanden:

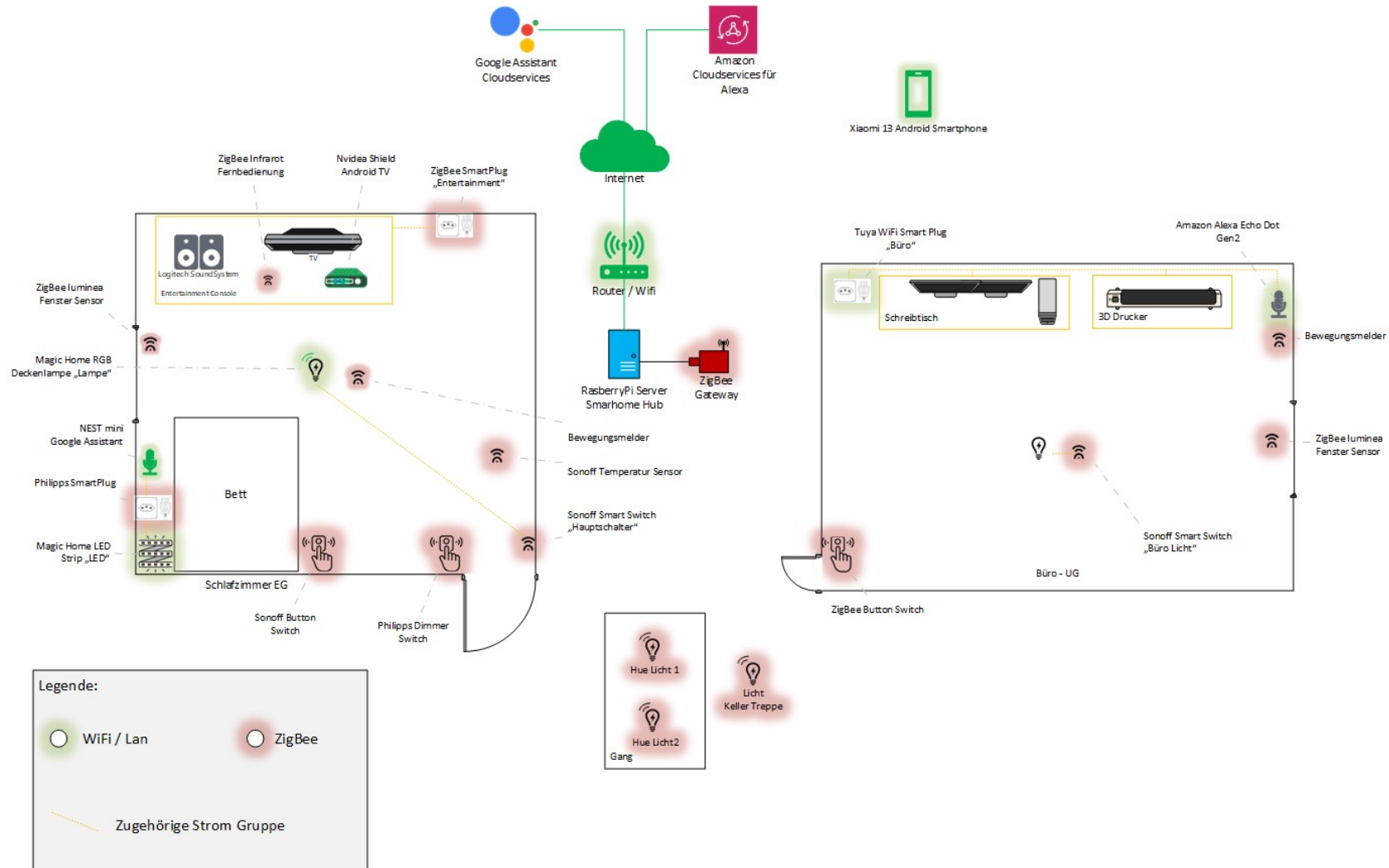








Abbildung 7 Detail Architektur Ansicht

### 6.4.3 Verwendete Geräte

In der Vorbereitung für diese Arbeit wurden einige Produkte auf Aliexpress bestellt. Nachfolgend sind alle selektierten Produkte aufgelistet – mit einer Preisangabe, der genutzten Technologie und der Angabe dessen, ob sie verwendet wurden.

#### Aliexpress-Geräte:

Produktbild	Produktname	Technologie	Anzahl	Preis	Verwendet
	Tenky Zigbee Smart MINI Bewegungsmelder	Zigbee	1	10.68 \$	ja
	Moes ZigBee Smart Stecker mit 2 USB	Zigbee	1	21.60 \$	ja
	Tuya Wifi Smart Socket USB	WiFi	1	13.50 \$	ja
	IHSENO ZigBee PIR Motion Sensor	Zigbee	1	15.43 \$	ja
	16A WiFi Smart Steckdose Tuya Smart Life	WiFi	1	12.13 \$	nein, falsches Produkt
	Sonoff zigbee 3,0 usb Dongle Gateway	Zigbee Gateway	1	19.90 \$	ja

 <p><b>SNZB-01 Button Switch</b></p>	<p>Sonoff Zigbee SNZB-01 Button Switch</p>	<p>Zig- bee</p>	<p>1</p>	<p>7.99 \$</p>	<p>ja</p>
 <p><b>SNZB-02 Tem. &amp; Hum. Sensor</b></p>	<p>Sonoff SNZB-02 Temperature and Humidity Sensor</p>	<p>Zig- bee</p>	<p>1</p>	<p>8.49 \$</p>	<p>ja, wurde re- pariert</p>
	<p>Smart ZigBee Ir Fernbedienung Tuya Smart Home Fernbedie- nung</p>	<p>Zig- bee</p>	<p>1</p>	<p>13.35 \$</p>	<p>ja</p>
	<p>SONOFF NSPanel Smart Szene Wand Schalter eWelink</p>	<p>WiFi</p>	<p>1</p>	<p>64.90 \$</p>	<p>nein</p>
	<p>Excellux Tuya Zigbee Led Smart Glüh- birne E27</p>	<p>Zig- bee</p>	<p>4</p>	<p>27.99 \$; 6.99 \$ einzeln</p>	<p>ja</p>
	<p>AUBESS 16A Tuya WiFi Mini DIY Smart Switch</p>	<p>WiFi</p>	<p>3</p>	<p>10.85 \$; 2.72 \$ einzeln</p>	<p>nein</p>

Tabelle 4 Übersicht der Smart Devices von Aliexpress

### Digitec-Bestellung

Am Anfang des Projektes wurden auf Digitec weitere Sensoren eingekauft, welche nachfolgend aufgelistet werden:

Produktbild	Produktname	Technologie	Anzahl	Preis	Verwendet
	Sonoff Zig-Bee Smart Switch	Zigbee	3	24.45 CHF	ja, nur 1 Stück
	Luminea ZigBee Door & Window Alarm	Zigbee	2	57.90 CHF	ja
	Philips Hue White BT	Zigbee/Nest	2	41.90 CHF	ja
	Philips Hue SmartPlug CH	Zigbee	1	39.90 CHF	ja
	Philips Hue Dimmer Switch 2021	Zigbee	1	18.90 CHF	ja

Tabelle 5 Übersicht der Smart Devices von Digitec

Für den Home-Assistant-Server wurde ein Raspberry Pi 4 verwendet, welcher schon in meinem Besitz war und der somit nicht neu bestellt wurde. Sein Originalpreis betrug im Jahr 2020 89 CHF. Einige der aufgelisteten Geräte wurden nicht eingesetzt, weil entweder die Technologie nicht genutzt werden sollte oder ein defekter bzw. falscher Sensor geliefert wurde. Andere Gründe, aus denen Geräte nicht genutzt wurden, werden in den weiteren Teilen der Arbeit beschrieben. Es folgen verwendete Geräte, die ich schon besass und somit nicht neu gekauft habe.

Produktbild	Produktname	Technologie	Anzahl	Preis	Verwendet
	Google Nest Mini	WiFi	1	37 CHF	ja
	Amazon Echo Dot Gen2	WiFi	1	gekauft für 49.90 CHF	ja
	Offdarks Magic Home LED Deckenlicht	WiFi	1	29.58 \$	ja
	Magic Home Neon LED Strip Blue	WiFi	1	26.99 \$	ja

Tabelle 6 Vorhandene Produkte

Alle oben angegebenen Geräte sind mittlerweile günstiger erhältlich und von den Sprachassistenten gibt es neuere Generationen sowie neue Gerätetypen. Die Produkte sollten dennoch kompatibel sein.

## 7 Umsetzung des Smart-Home-Projekts

In diesem Kapitel geht es um die praktische Umsetzung des Projekts. Aufgebaut ist dieses durch eine Einführung zum Home Assistant Server mit anschliessender Anleitung und Zusammenfassung der Umsetzung.

### 7.1 Funktionen von Home Assistant

Um den weiteren Verlauf der Umsetzung in den nachfolgenden Kapiteln verständlich zu machen, werden zunächst die Grundfunktionen von Home Assistant erläutert. Das System besteht aus verschiedenen Komponenten.

#### Dashboard

Dashboards sind anpassbare Seiten zum Anzeigen von Informationen in Home Assistant. Standardmässig gibt es zwei Dashboards: Übersicht und Energie. Das Übersichts-Dashboard ist die erste Seite, die nach dem Onboarding-Prozess angezeigt wird. In Abbildung 8 wird ein individualisiertes Beispiel des Übersichts-Dashboards vorgestellt. Wenn der Benutzer das Onboarding neu vorgenommen hat, ist diese Seite fast leer. Weitere und genauere Informationen zu den Dashboards sind auf der Webseite von Home Assistant einsehbar: <https://www.home-assistant.io/dashboards>.

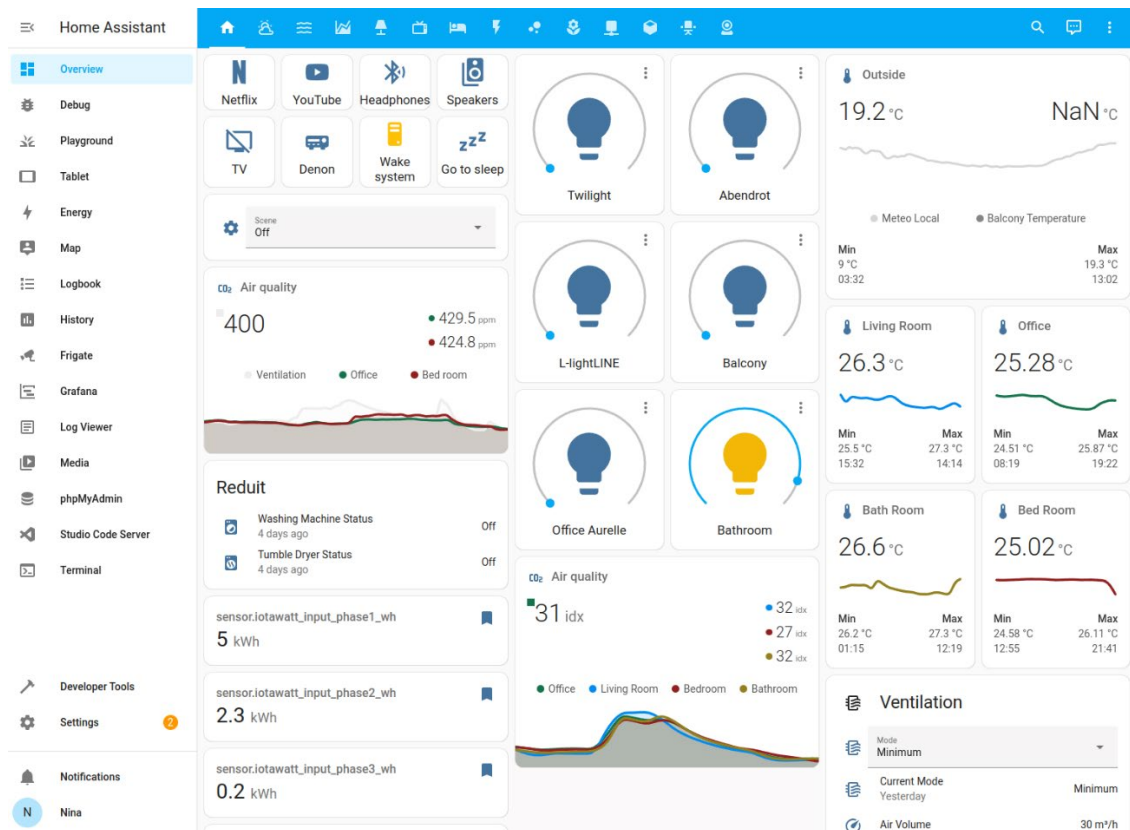


Abbildung 8 Beispiel der Dashboard-Ansicht

## Integrations

Integrationen sind Softwarekomponenten, die es Home Assistant ermöglichen, sich mit anderen Softwareanwendungen und Plattformen zu verbinden. Zum Beispiel wurde ein Produkt von Philips namens Hue die Philips Hue Integration verwendet, damit Home Assistant mit dem Hardware-Controller Hue Bridge kommunizieren kann. Alle mit Home Assistant kompatiblen Geräte, die an die Hue Bridge angeschlossen sind, erscheinen in Home Assistant.

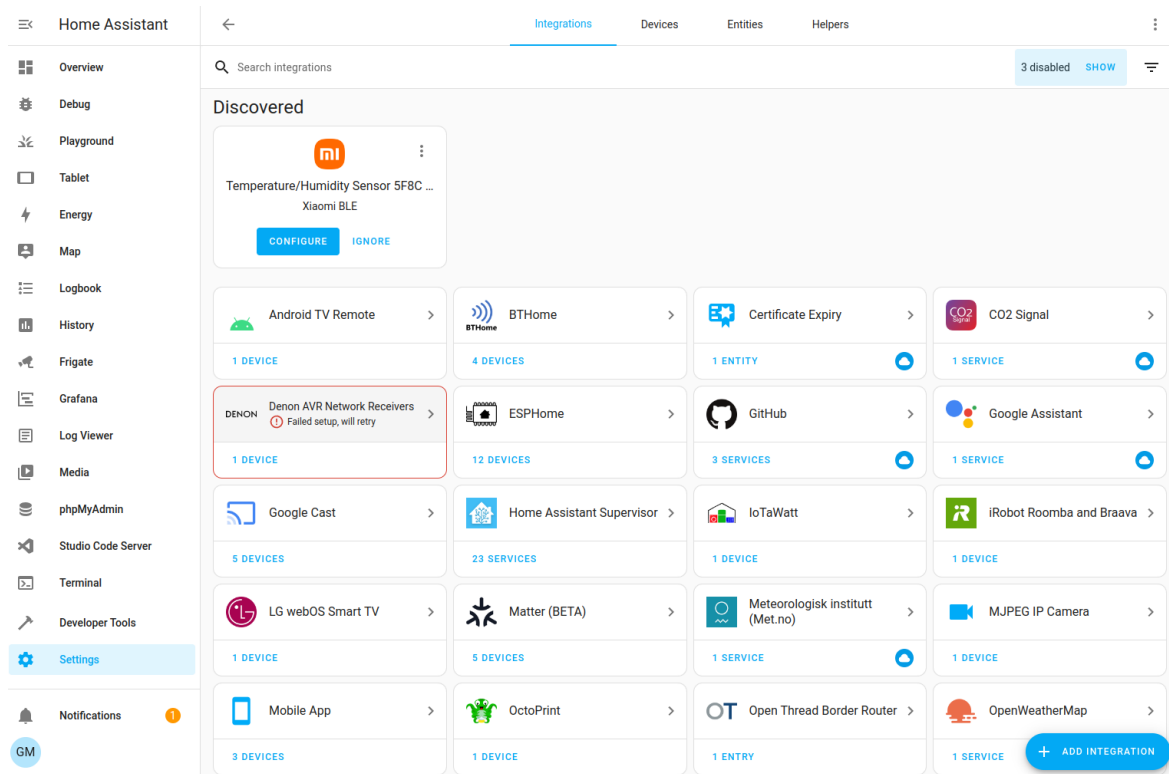


Abbildung 9 Vorschau für Integrationen

## Geräte & Entitäten

Geräte sind eine logische Gruppierung einer oder mehrerer Entitäten. Ein Gerät kann ein physisches Gerät repräsentieren, das ein oder mehrere Sensoren haben kann. Die Sensoren (zum Beispiel Bewegungssensoren) werden als mit dem Gerät verbundene Entitäten dargestellt. Mit ihnen können die Bewegungserkennung, die Temperatur und die Lichtstärke als Entitäten bereitgestellt werden. Entitäten weisen Zustände wie <erkannt> auf, wenn Bewegung erkannt wird, und <clear>, wenn keine Bewegung festgestellt wird.

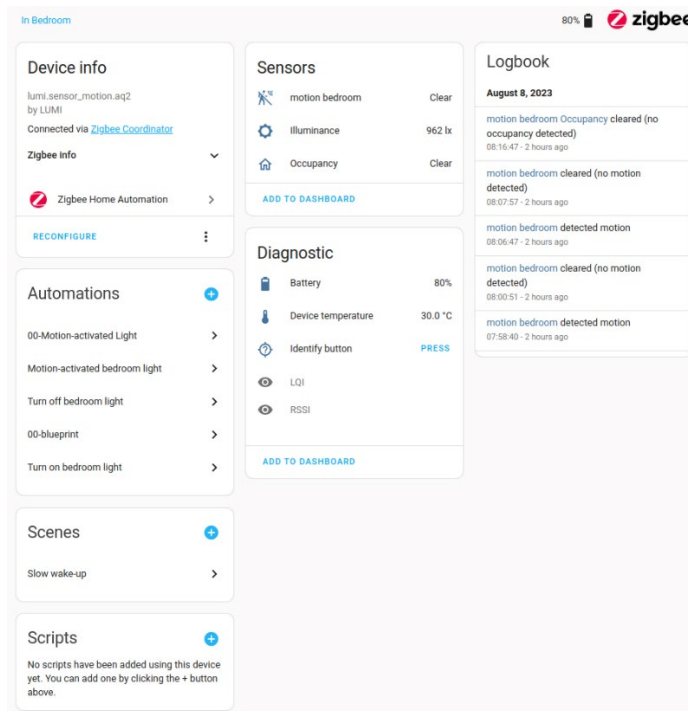


Abbildung 10 Vorschau der Geräte- und Entitätenkonfiguration

## Automatisierungen

Hierbei handelt es sich um eine Gruppe wiederholbarer Aktionen, die automatisch ausgeführt werden können. Automatisierungen bestehen aus den nachfolgenden drei Schlüsselkomponenten.

- **Auslöser:** Ereignisse, die eine Automatisierung starten, zum Beispiel das Einstellen der Sonne
- **Bedingungen:** optionale Tests, die erfüllt sein müssen, bevor Aktionen ausgeführt werden (zum Beispiel wenn jemand zu Hause ist)
- **Aktionen:** Interagieren mit Geräten, zum Beispiel beim Einschalten einer Lampe

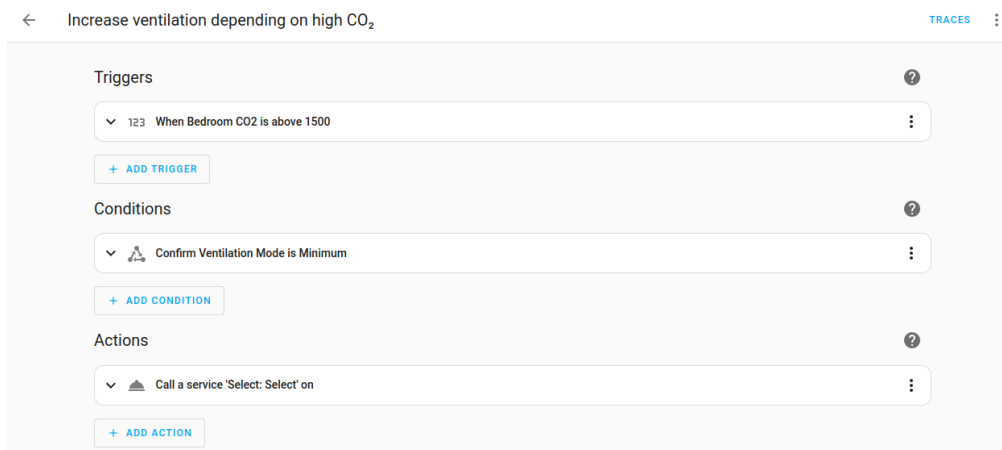


Abbildung 11 GUI für die Automationserstellung

(GUI steht für Graphical User Interface)

## Skripte

Ähnlich wie Automatisierungen sind Skripte wiederholbare Aktionen, die ausgeführt werden können. Der Unterschied zwischen Skripten und Automatisierungen besteht darin, dass Skripte keine Auslöser wie Automatisierungen haben. Das bedeutet, dass Skripte nicht automatisch ausgeführt

werden können, es sei denn, sie werden in einer Automatisierung verwendet. Skripte sind besonders nützlich, wenn die gleichen Aktionen in verschiedenen Automatisierungen ausgeführt oder von einem Dashboard aus eingeleitet werden sollen. Informationen zur Erstellung von Skripten befinden sich in der Skript-Dokumentation.

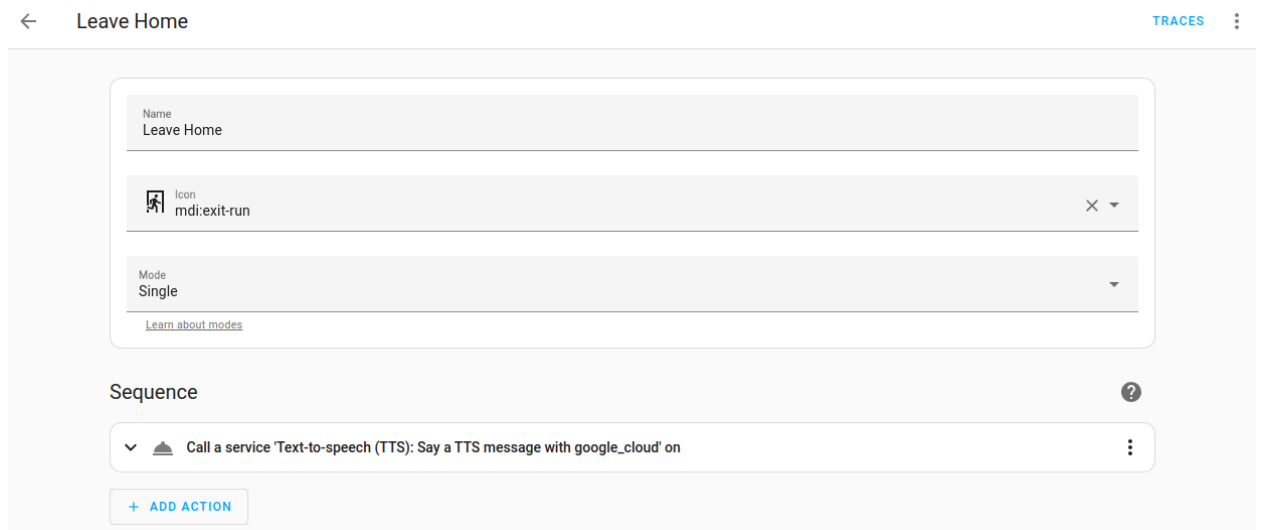


Abbildung 12 Vorschau der Skriptkonfiguration

### **Vorlagen/Blueprints**

Eine Vorlage ist eine Skript- oder Automatisierungskonfiguration mit bestimmten Teilen, die als konfigurierbar markiert sind. Sie ermöglicht es, verschiedene Skripte oder Automatisierungen auf der Grundlage derselben Vorlage zu erstellen.

Zum Beispiel möchte ein Nutzer Lichter auf der Grundlage von Bewegung steuern. Eine Vorlage stellt das generische Automatisierungsframework bereit und ermöglicht es ihm, einen bestimmten Bewegungssensor als Auslöser auszuwählen sowie das Licht genau einzustellen. Diese Vorlage kann somit zwei Automatisierungen erstellen. Jede Automatisierung verfügt über ihre eigene Konfiguration und arbeitet unabhängig von der anderen. Dennoch teilen sie sich einige grundlegende Automatisierungskonfigurationen, um diese nicht doppelt einrichten zu müssen.

Vorlagen werden von der Community im Vorlagen-Community-Forum geteilt.

### **Szenen**

Szenen ermöglichen es Nutzern, vordefinierte Einstellungen für ihre Geräte zu erstellen. Ähnlich wie ein Fahrmodus auf Mobiltelefonen oder wie Fahrerprofile in Autos können Szenen die Umgebung individuellen Bedürfnissen anpassen. Zum Beispiel kann die Szene 'Filme schauen' die Beleuchtung dimmen, den Fernseher einschalten und die Lautstärke erhöhen. Diese Konfiguration kann als Szene gespeichert und verwendet werden, ohne jedes Mal einzelne Geräte einzustellen. Die Szenendokumentation enthält weitere Informationen zur Nutzung von Szenen.



Abbildung 13 Vorschau des Szenen-Editor

### Add-ons

Abhängig von ihrem Installationstyp können Nutzer Erweiterungen von Drittanbietern installieren. Erweiterungen sind in der Regel Apps, die mit Home Assistant ausgeführt werden können, aber eine schnelle und einfache Möglichkeit aufweisen, sie in Home Assistant zu installieren, zu konfigurieren und auszuführen. Über Erweiterungen werden zusätzliche Funktionen erlangt, während Integrationen Home Assistant mit anderen Apps verbinden.

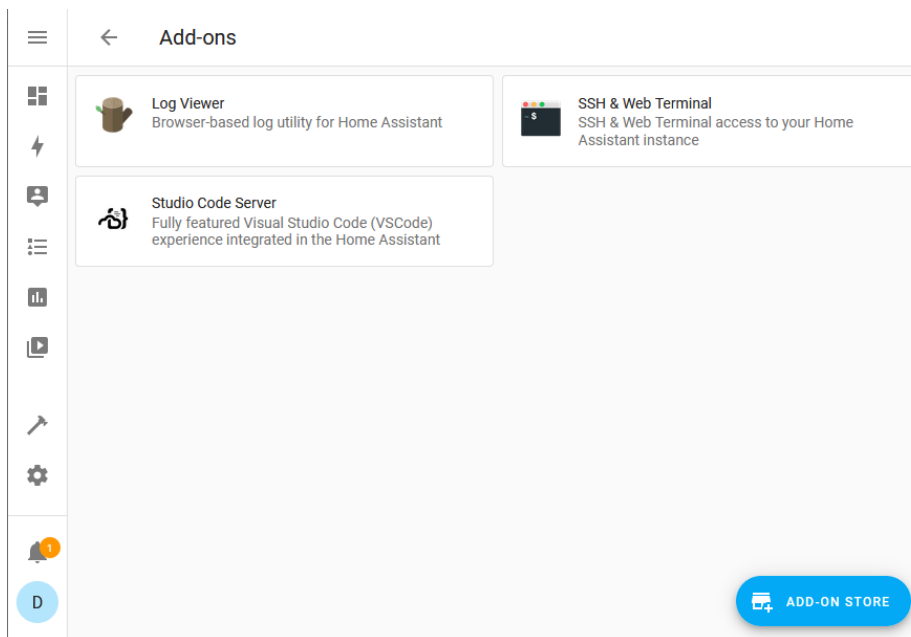


Abbildung 14 Vorschau der Add-on-Übersicht

Die Umsetzung ist mehrheitlich als Anleitung geschrieben.

Entscheidungen und Informationen sind direkt ergänzt und zum Teil markiert, um ihre Relevanz hervorzuheben.

## 7.2 Aufbau von Home Assistant

Auf der Plattform Home Assistant werden vier verschiedene Installationsmethoden angeboten. Zu empfehlen ist die Verwendung einer der folgenden beiden Lösungen:

1. **Home Assistant Betriebssystem:** Es handelt sich um ein minimales Betriebssystem, das speziell für Home Assistant optimiert ist. Es wird mit Supervisor geliefert, um Home Assistant Core und Add-ons zu verwalten. Dies ist die empfohlene Installationsmethode.
2. **Home Assistant Container:** Ermöglicht wird die eigenständige Container-basierte Installation von Home Assistant Core (z. B. Docker).

Es gibt zwei alternative Installationsmethoden für erfahrene Benutzer:

3. **Home Assistant Supervised:** manuelle Installation des Supervisors
4. **Home Assistant Core:** manuelle Installation unter Verwendung einer Python Virtual Environment

Vergleich der Installationsmethoden mit einer Feature-List:

	OS	Container	Core	Supervised
<a href="#">Automations</a>	✓	✓	✓	✓
<a href="#">Dashboards</a>	✓	✓	✓	✓
<a href="#">Integrations</a>	✓	✓	✓	✓
<a href="#">Blueprints</a>	✓	✓	✓	✓
Uses container	✓	✓	✗	✓
<a href="#">Supervisor</a>	✓	✗	✗	✓
<a href="#">Add-ons</a>	✓	✗	✗	✓
<a href="#">Backups</a>	✓	✓ <sup>1</sup>	✓ <sup>1</sup>	✓
Managed Restore	✓	✗ <sup>2</sup>	✗ <sup>2</sup>	✓
Managed OS	✓	✗	✗	✗

1: Backups for Home Assistant Core and Home Assistant Container is provided by the [backup integration](#). 2: Backups for Home Assistant Core and Home Assistant Container are either a tool to migrate to HAOS or a completely manual restore of the backup.

Abbildung 15 Home Assistant Feature-List

Da ein Raspberry Pi verwendet wurde, war die OS(Operating System)-Installation die beste und einfachste Wahl.

Das Image für den Pi wurde mittels Pi Imager installiert. Andere Installationsmöglichkeiten sind unter <https://www.home-assistant.io/installation/> einsehbar. Es wurde folgende Anleitung befolgt:

- Lade den Raspberry Pi Imager auf deinen Computer herunter und installiere ihn, wie unter <https://www.raspberrypi.com/software/> beschrieben.
- Öffne den Raspberry Pi Imager.
- Wähle das Betriebssystem aus:
  - Wähle <Betriebssystem auswählen>.
  - Wähle <Andere spezielle Zweckbetriebssysteme > Home Assistants und Home Automation > Home Assistant>.

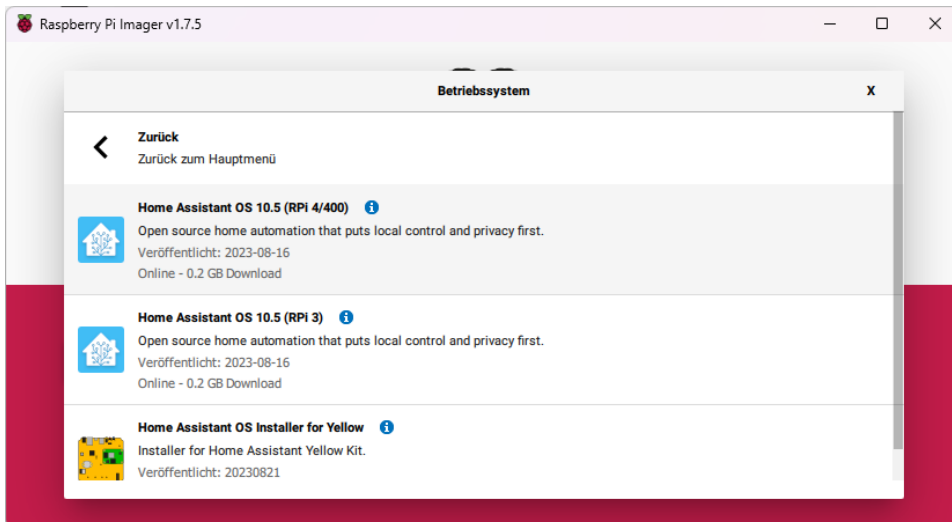


Abbildung 16 Pi-Imager-OS-Auswahl

- Wähle das Home-Assistant-Betriebssystem aus, das zu deiner Hardware passt (RPi 3 oder RPi 4).
- Wähle den Speicherort:
  - Stecke die SD(Secure Digital Memory)-Karte in den Computer.  
Hinweis: Der Inhalt der Karte wird überschrieben.  
Wähle deine SD-Karte aus.



Abbildung 17 Pi Imager – Schreiben des OS auf die SD-Karte

- Schreibe den Installer auf die SD-Karte:
  - Um den Vorgang zu starten, wähle <Schreiben>.
  - Warte, bis das Home-Assistant-Betriebssystem auf die SD-Karte geschrieben wurde.
- Werfe die SD-Karte aus.

Starten des Raspberry Pi:

1. Stecke die SD-Karte in deinen Raspberry Pi.
2. Verbinde ein Ethernet-Kabel mit dem Netzwerk.
3. Schliesse das Netzteil an, um das Gerät zu starten.

4. Öffne in deinem Desktop-Webbrowser innerhalb weniger Minuten deine neue Home-Assistent-Installation unter <http://homeassistant.local:8123>.
5. Hinweis: Bei meiner Installation habe ich zwölf Minuten gewartet. Somit bitte genug Zeit einrechnen und geduldig bleiben.

Vorbereitung von Home Assistant:

Nachdem Home Assistant installiert war, wurde das erste Set-up durchgeführt. In diesem Schritt galt es das Benutzer-Log-in von Home Assistant zu erstellen. Dieses Konto ist ein Administrator-Konto und hat die Berechtigung, alles zu ändern. Es wurde folgende Anleitung befolgt:

Wenn eine Wiederherstellung aus dem Backup einer früheren Installation gewünscht ist, wähle **Wiederherstellung aus Backup**.

Fahre mit dem Verfahren zur Wiederherstellung aus dem Backup fort, sollte bereits eine Konfiguration vorhanden sein.

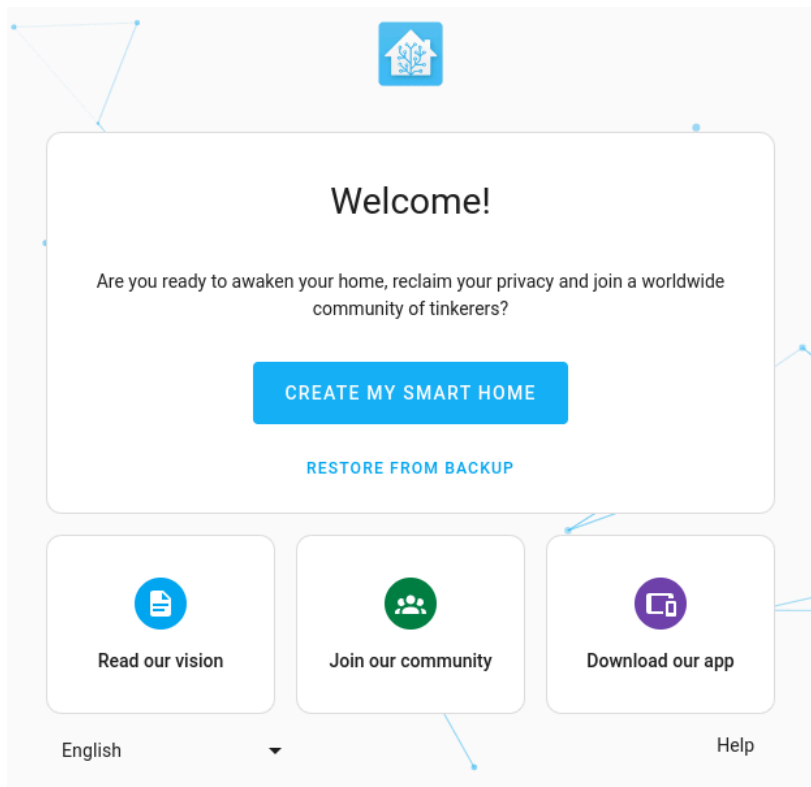


Abbildung 18 Vorschau des Willkommens-Bildschirms

Mit den nachfolgenden Schritten wird beschrieben, wie eine neue Installation erstellt werden kann.

- Wähle **Mein Smart Home erstellen**.
- Gib einen Namen, einen Benutzernamen und ein Passwort ein. Wähle **Konto erstellen**.

- Lege den Benutzernamen und das Passwort fest.

Abbildung 19 User-Set-up

- Gib den Standort des Zuhauses ein.

Hinweis:

Der Standort wird verwendet, um Einstellungen wie Zeitzonen, Einheitensysteme und Währungen zu konfigurieren. Er wird auch für standortbasierte Informationen und Automatisierungen genutzt, z. B. zur Anzeige der Wettervorhersage, zum Öffnen der Jalousien bei Sonnenaufgang oder zum Starten des Staubsaugers beim Verlassen des Hauses. Wenn der Nutzer den Standort nicht teilen möchte, kann er einen Ort weit entfernt von seinem tatsächlichen Wohnort auswählen. Er kann diese Information jederzeit in den Einstellungen ändern.

- Wähle aus, welche Informationen geteilt werden sollen.  
Hinweis: Standardmässig ist die Freigabe deaktiviert, dies kann aber jederzeit geändert werden.

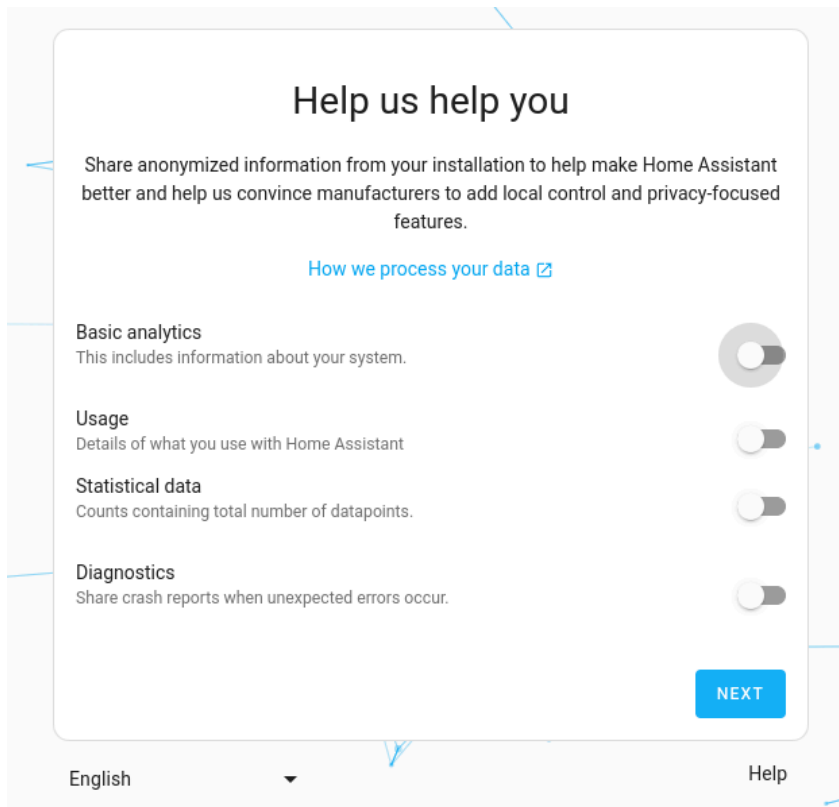


Abbildung 20 Informationsteilung

- Wähle <Weiter>, sobald die Schritte durchgeführt wurden.

Home Assistant zeigt anschliessend alle Geräte an, die es im Netzwerk entdeckt hat. Es ist unproblematisch, wenn weniger Geräte als erwartet angezeigt werden, denn Nutzer können später jederzeit manuell Geräte hinzufügen.

Schliesslich sollte <Fertig> angeklickt werden.

Nun gelangt der Nutzer zur Home-Assistant-Web-Oberfläche. Geräte, die automatisch erkannt und eingerichtet wurden, kann dieses Standard-Dashboard bereits anzeigen.

### 7.3 Grundkonfiguration von Home Assistant

Nach der Basisinstallation wurden einige Grundkonfigurationen getätigt.

Als Erstes wurde der Home Assistant mit den neusten Updates versorgt.

Danach wurde der Sonoff Zigbee Gateway Stick angeschlossen, welcher ohne Probleme erkannt wurde.

Zudem wurde der Router auch automatisch erkannt, aber für die Nutzung seiner Funktionen wurde ein Log-in benötigt. Dieses musste auf dem Router erstellt werden. Ich besitze aktuell eine Fritz!Box 7490.

Dort konnte einfach unter System und Fritz!Box-Benutzer ein User angelegt werden:

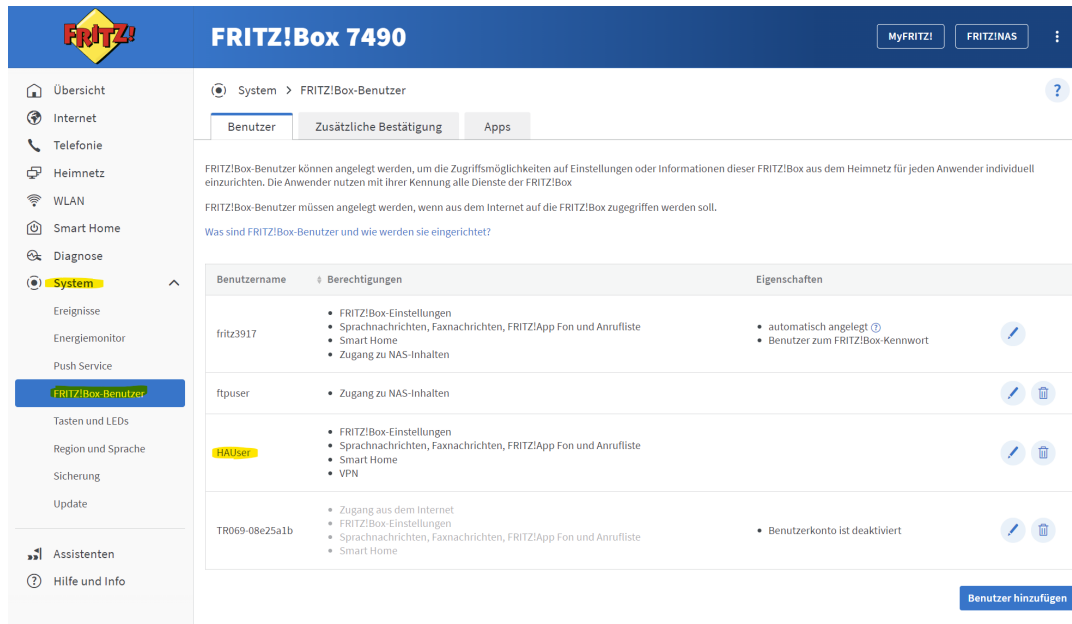


Abbildung 21 Fritz!Box-User-Set-up

Dieses Log-in wurde bei HA(Home Assistant) eingetragen (der Port wurde auf dem Standard belassen):

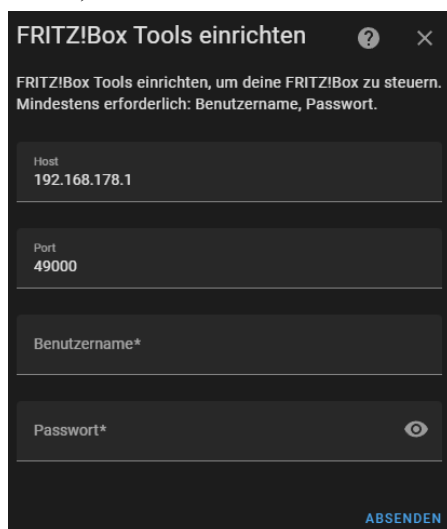


Abbildung 22 Konfiguration des HA-Zugriffs auf die Fritz!Box

Mit diesen Einstellungen wurden alle WLAN-fähigen Geräte erkannt, so konnten sie für Automationen genutzt werden. Zudem wurden weitere Geräte, genauer Integrationen wie Google Chrome Cast, Android TV Remote und EufyLife, zu Home Assistant hinzugefügt, ohne dass eine gezielte Konfiguration notwendig war.

Als Nächstes wurde die Umgebung zum Internet exposed. Dies bedeutet, dass eine Portöffnung auf dem Router eingeleitet wurde, um direkt aus dem Internet auf den Raspberry Pi zugreifen zu können. Dieser Schritt war nötig, um später Google Assistent und Amazon Alexa verbinden zu können, was einem der Ziele entsprach. Dazu wurde zunächst auf dem Router ein Port-Forwarding auf den Standard-Port von HA und zusätzlich zu 443 erstellt.

Die genutzte Oberfläche der Fritz!Box ist in Abbildung 19 einsehbar. Bei anderen Routern sollte die Methode im Handbuch nachgelesen werden.

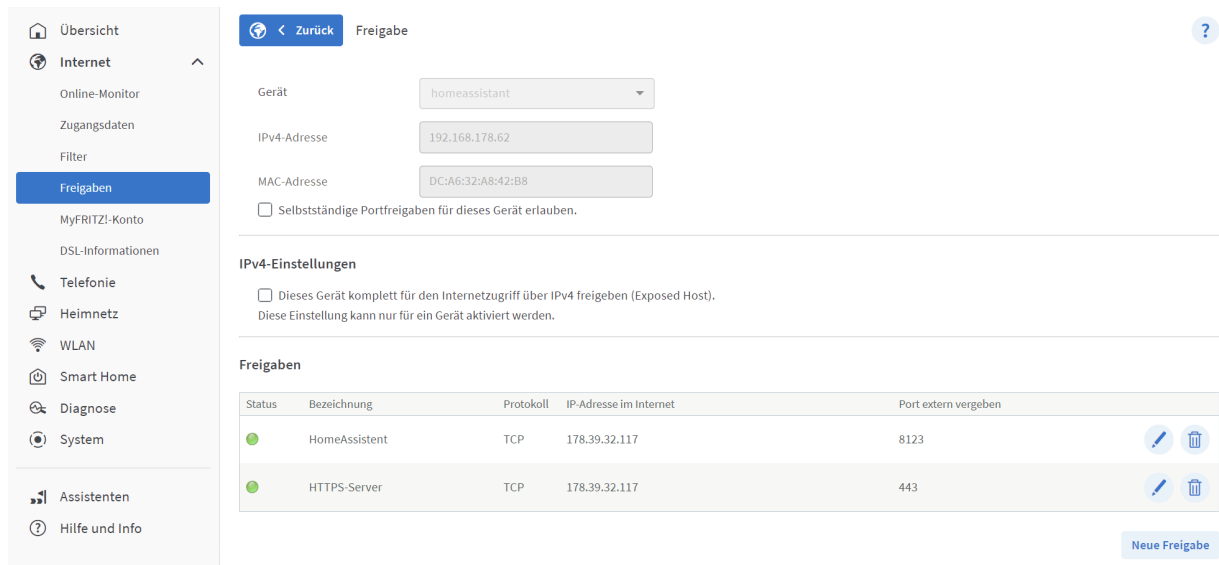


Abbildung 23 Fritz!Box-Port-Konfiguration

Anschliessend habe ich für die erleichterte Verwaltung von Konfigurationen und zum Absichern meiner Instanz mittels eines SSL(Secure Sockets Layer)-Zertifikats folgende Add-ons zu Home Assistant hinzugefügt: «Advanced SSH(Secure Shell) & Webterminal», «File Editor» und «Duck DNS» (DNS steht für Domain Name System).

Diese konnten unter Einstellungen – Add-ons – Add-on Store installiert werden.

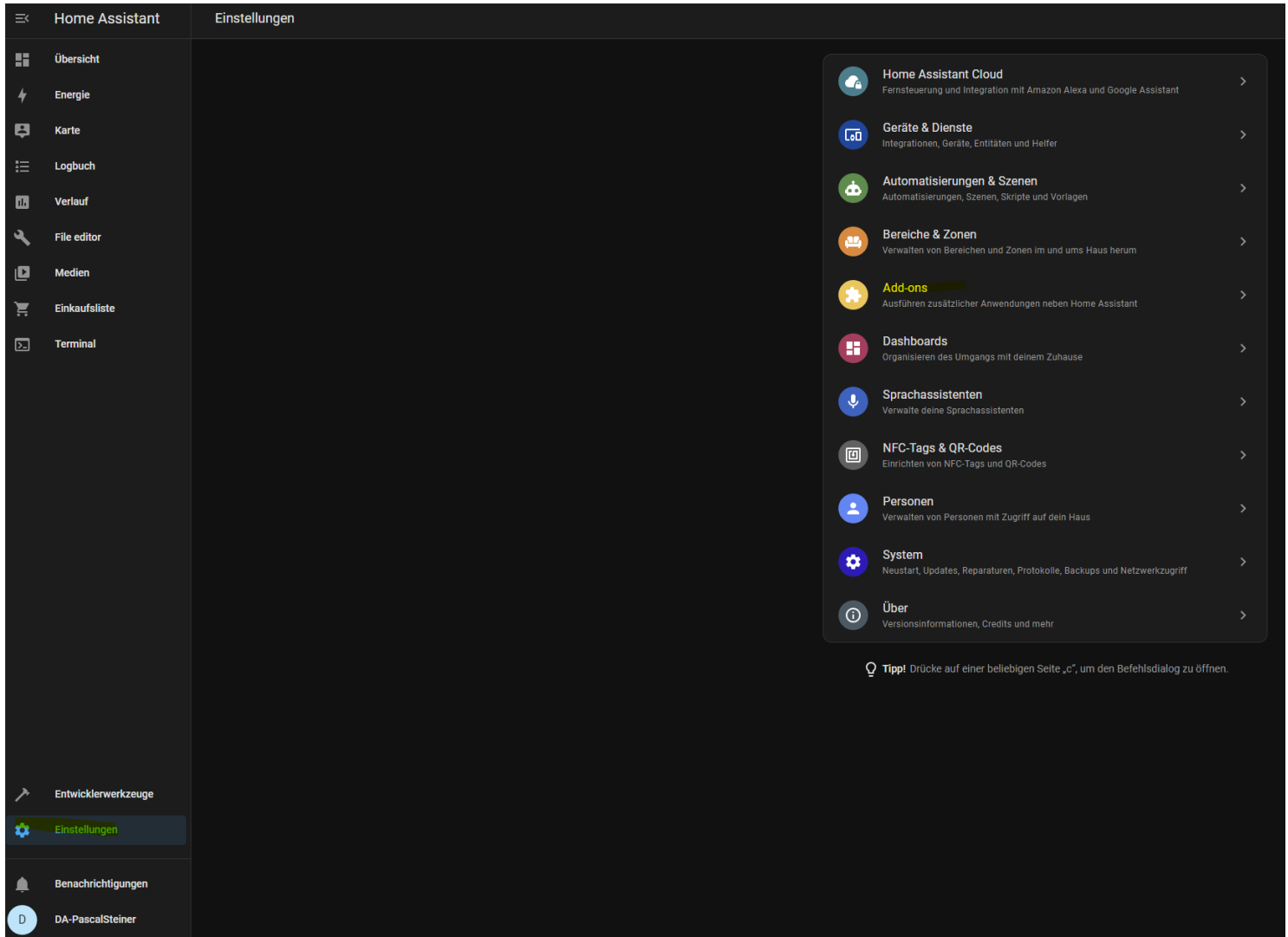


Abbildung 24 Weg zum Add-on-Store

Der Add-on-Store wurde nach Klick auf den Button unten rechts («Add-on-Store öffnen») ange-  
zeigt.

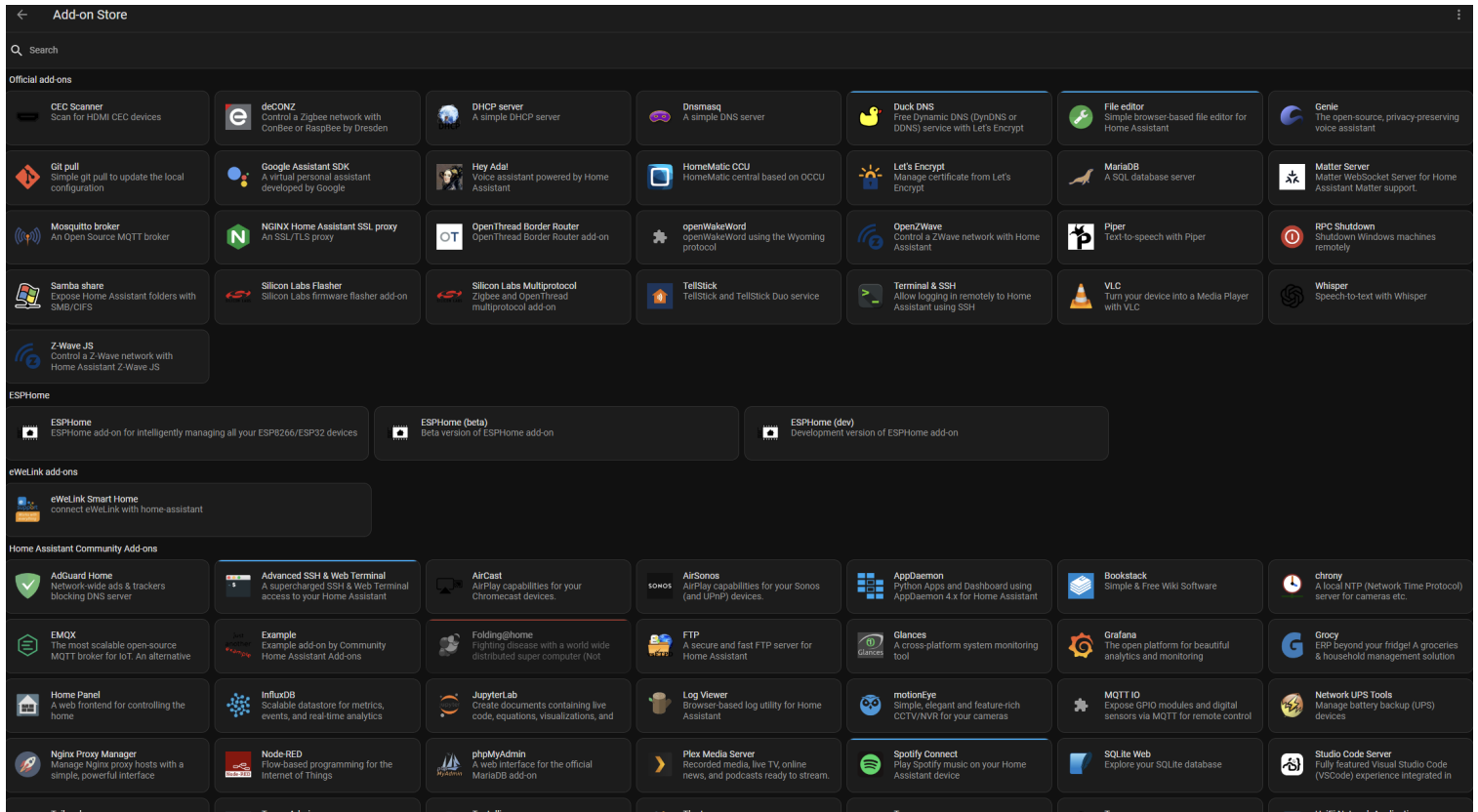


Abbildung 25 Add-on-Store-Übersicht

Nach der Installation musste DuckDNS konfiguriert werden, bevor das Add-on gestartet werden konnte.

The screenshot displays the configuration interface for the DuckDNS add-on. At the top, there are navigation tabs: 'Informationen', 'Dokumentation', 'Konfiguration' (selected), and 'Protokoll'. The main content area is titled 'Duck DNS' and shows the current version as 1.15.0. A 'Bewertung' (rating) of 5 is displayed. Below this, there is a description of the service and a link to the DuckDNS website. The configuration section includes several toggle switches and progress bars: 'Beim Booten starten' (checked), 'Watchdog' (checked), 'Automatische Updates' (checked), 'Hostname' (set to 'core-duckdns'), 'CPU-Auslastung des Add-ons' (0%), and 'RAM-Auslastung des Add-ons' (0.3%). At the bottom of the configuration section, there are buttons for 'STOPPEN', 'NEU STARTEN', and 'DEINSTALLIEREN'. Below the configuration section, there is a 'Home Assistant Add-on: DuckDNS' section with a description and a list of supported architectures: aarch64, amd64, armhf, armv7, and i386, all marked as 'yes'. An 'About' section provides further details about the service and the add-on's functionality.

Abbildung 26 Übersicht des DuckDNS-Add-on

Dazu musste wie in Abbildung 26 illustriert auf den Tab <Konfiguration> gewechselt werden, wo die nötigen Daten eingetragen wurden.

Hinweis: Für die meisten Add-ons ist es im Tab <Dokumentation> ersichtlich, wie sie zu konfigurieren sind. Im Fall von DuckDNS mussten beim Provider ein Account und eine Domain erstellt werden. Danach habe ich meine Domain und das Token in der Konfiguration eingetragen.

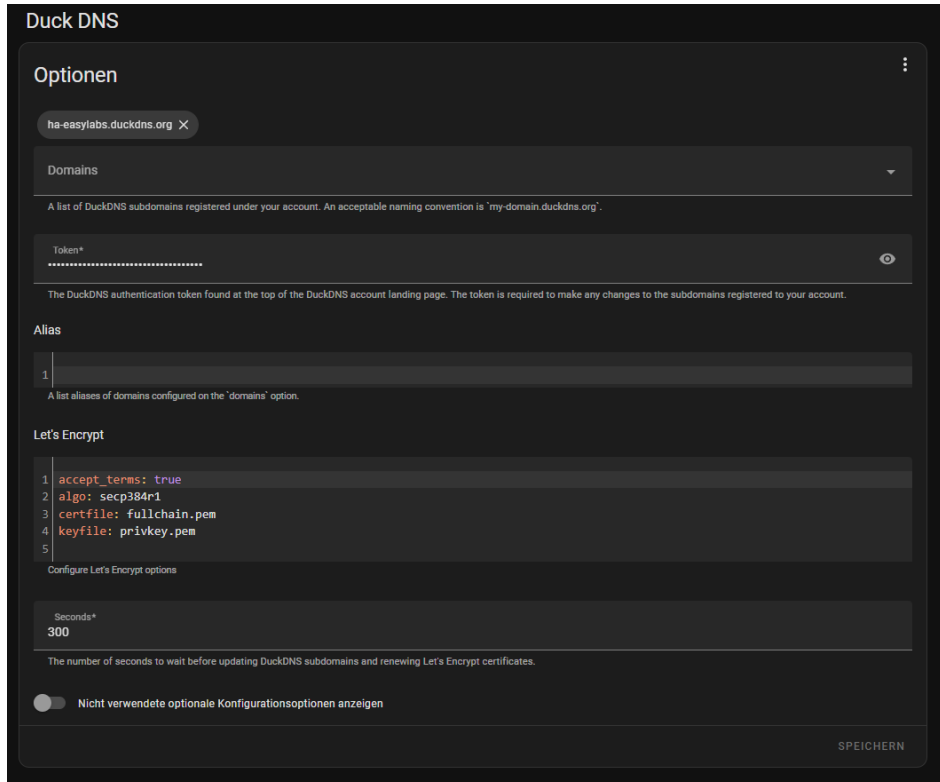


Abbildung 27 Konfiguration von DuckDNS

Somit konnte das Add-on im nächsten Schritt gestartet werden. Damit es bei einem Neustart jeweils ebenfalls aktiviert werden würde, habe ich <Watchdog> und <Beim Booten Starten> angeklickt und dann <Starten> gewählt.

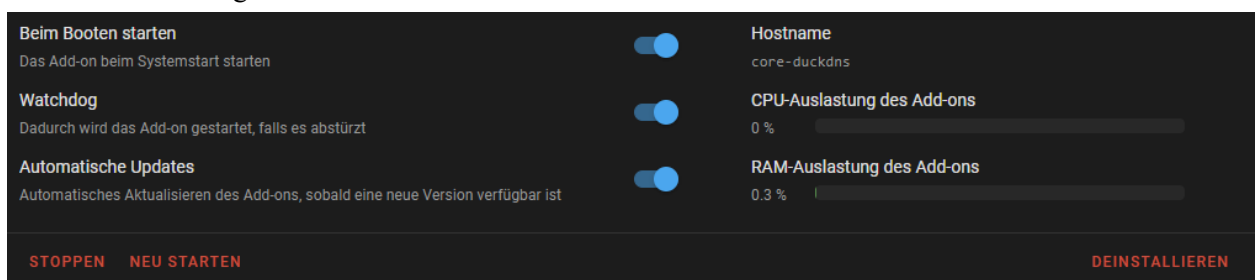


Abbildung 28 Starten des DuckDNS-Add-on

Im Ergebnis meldet sich Home Assistant selbstständig bei DuckDNS an und das Zertifikat wird automatisch generiert sowie erneuert. Nun musste nur noch Home Assistant konfiguriert werden, damit die Plattform das SSL-Zertifikat nutzt. Dazu wurde via File Editor <configuration.yaml> geöffnet und folgender Inhalt angefügt:

http:

```
ssl_certificate : /ssl/fullchain.pem
ssl_key : /ssl/privkey.pem.
```

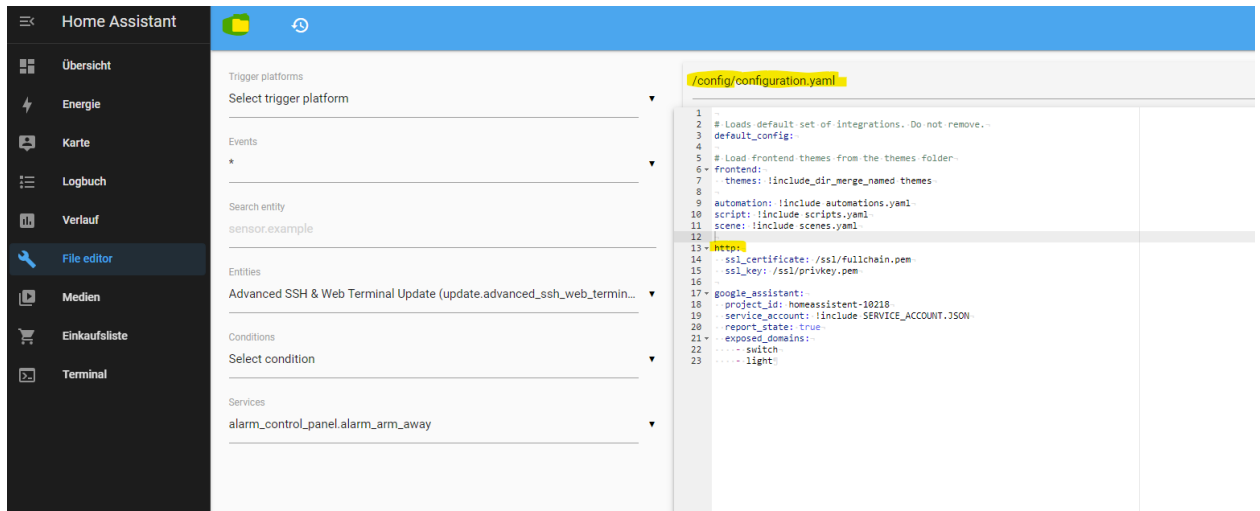


Abbildung 29 Konfigurationsdatei anpassen

Es galt das Speichern nicht zu vergessen und anschliessend musste Home Assistant komplett neu gestartet werden, da das erneute Laden der Konfiguration nicht gereicht hat.

Der Neustart der Instanz wurde wie folgt vorgenommen:

- gehe zu den Einstellungen
- klicke auf <System>
- klicke anschliessend oben rechts auf das Ausschalt-Symbol
- gehe im Popup auf <erweiterte Optionen> und dort auf <System neu starten>

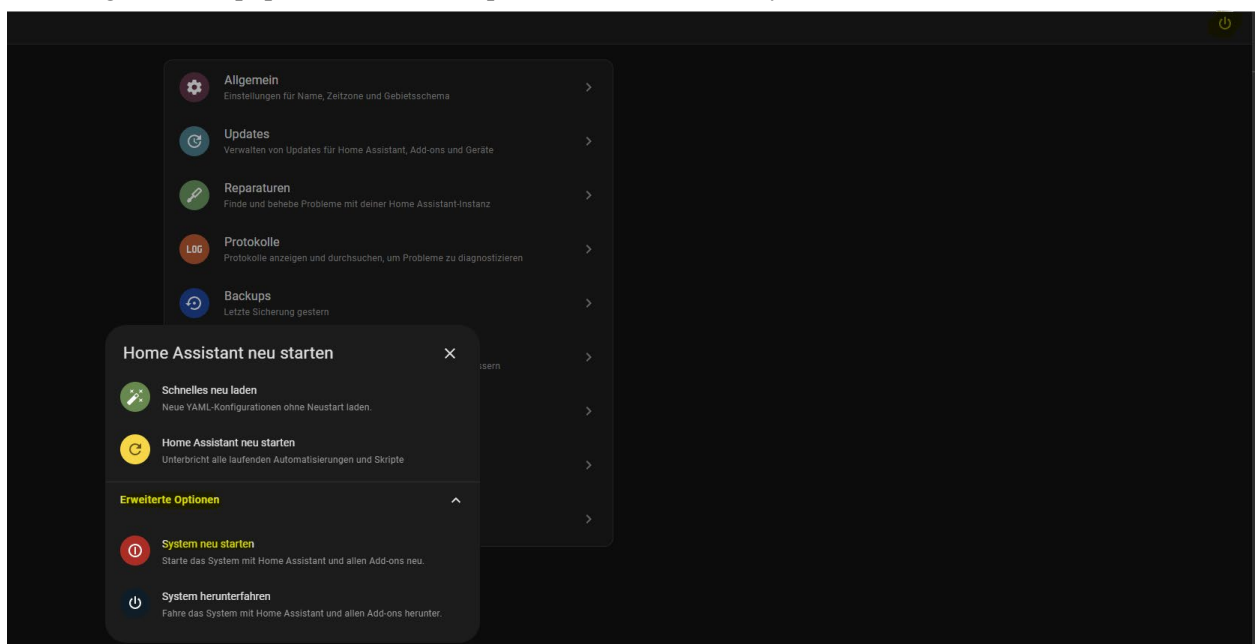


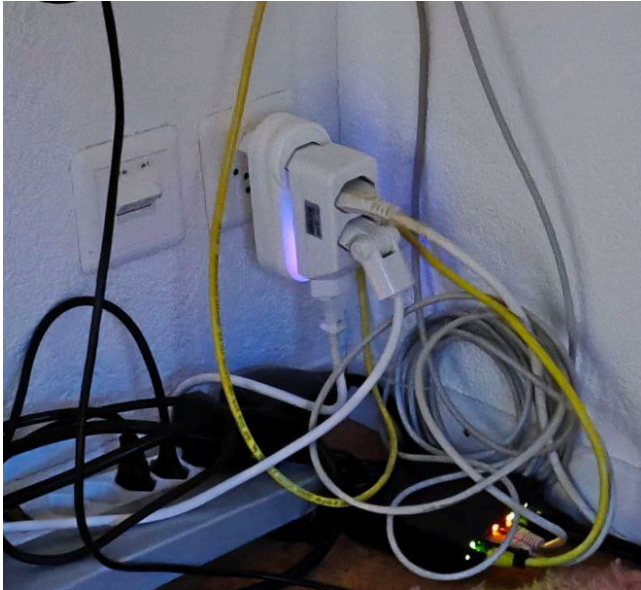
Abbildung 30 Neustart des HA-Systems

Anschliessend war der Home-Assistent-Server extern erreichbar und die Basiskonfiguration war abgeschlossen.

## 7.4 Smart-Geräte-Set-up

Die meisten Geräte sind relativ einfach zu installieren. Dazu müssen Nutzer den Stecker einstecken, die Lampen ersetzen und die Sensoren dort platzieren, wo sie eingesetzt werden sollen.

Meistens werden diese mit einem Kleber verkauft oder sie können angeschraubt bzw. einfach platziert werden. Nachfolgend sind Bilder meiner Installationen aufgeführt.



*Abbildung 31 Tuya Steckdose im Büro – Schreibtisch*



*Abbildung 32 Zigbee Steckdose für Entertainment – Schlafzimmer*

Beide Smart-Steckdosen sind bis 2600 Watt ausgelegt. Somit kann ohne Bedenken ein Mehrfachstecker verwendet werden, da die Last jeweils unter 1750 Watt bleibt.



Abbildung 33 Philipps Steckdose Nachttisch – Schlafzimmer



Abbildung 34 Tür- und Fenstersensor



Abbildung 35 Temperatursensor – Schlafzimmer



Abbildung 36 Magic Home Lampe und Zigbee Bewegungsmelder – Schlafzimmer



Abbildung 37 Bewegungsmelder im Büro

Bei den Geräten, welche eingebaut werden, sollte zunächst geprüft werden, ob genug Platz zur Verfügung steht. Ich wollte viele Steckdosen mit Einbau-Switches ausstatten, jedoch war nur bei einer Steckdose genug Platz vorhanden. Deswegen konnte ich gleich die smarten Steckdosen verwenden.

Es folgen Bilder dessen, wie ich trotzdem zwei Einbau-Switches installiert habe.

Jeder der nicht Strom affin ist sollte dies nicht selbst umsetzen!



Abbildung 38 Schalter vor der Demontage



*Abbildung 39 Eingebauter Sonoff Switch mit Änderung zu einfachem Stecker*



*Abbildung 40 Endresultat des Schalterumbaus*

Im Büro war leider der Platz hinter dem Schalter zu eng, somit konnte ich den Sonoff Smart Switch nicht direkt verbauen. Jedoch ist im Büro ein LED-Panel Verbaut welches nicht in die Decke eingelassen ist. Somit konnte der Smart Switch direkt bei der Lampe verbaut werden.

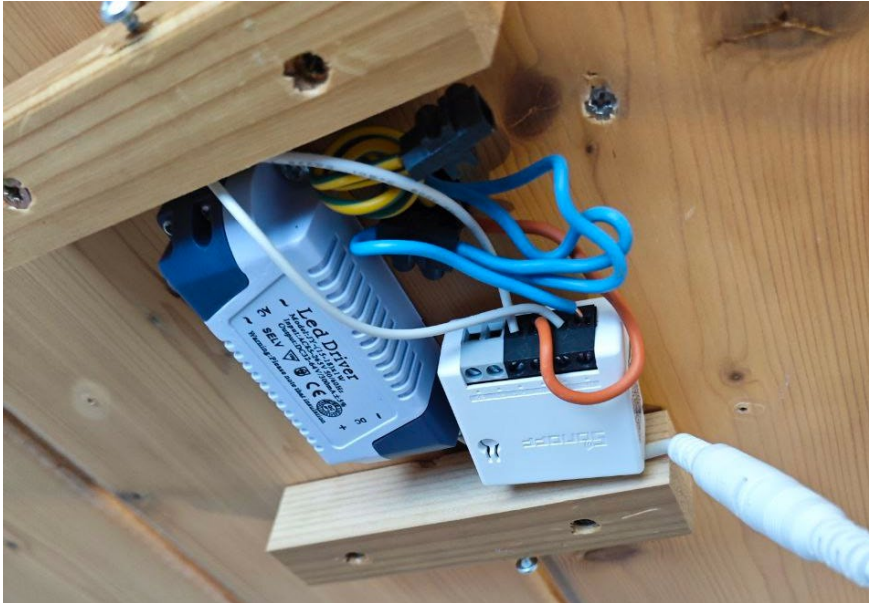


Abbildung 41 Smartswitch Einbau direkt bei der Lampe

## 7.5 Smart-Geräte-Integration

Zigbee-Integration:

Das Integrieren von Zigbee-Geräten ist einfach. Die Vorgehensweise wird nachfolgend beschrieben.

- Einstellungen aufrufen, dann «Geräte und Dienste» wählen sowie anschliessend «Zigbee Home Automation» anklicken

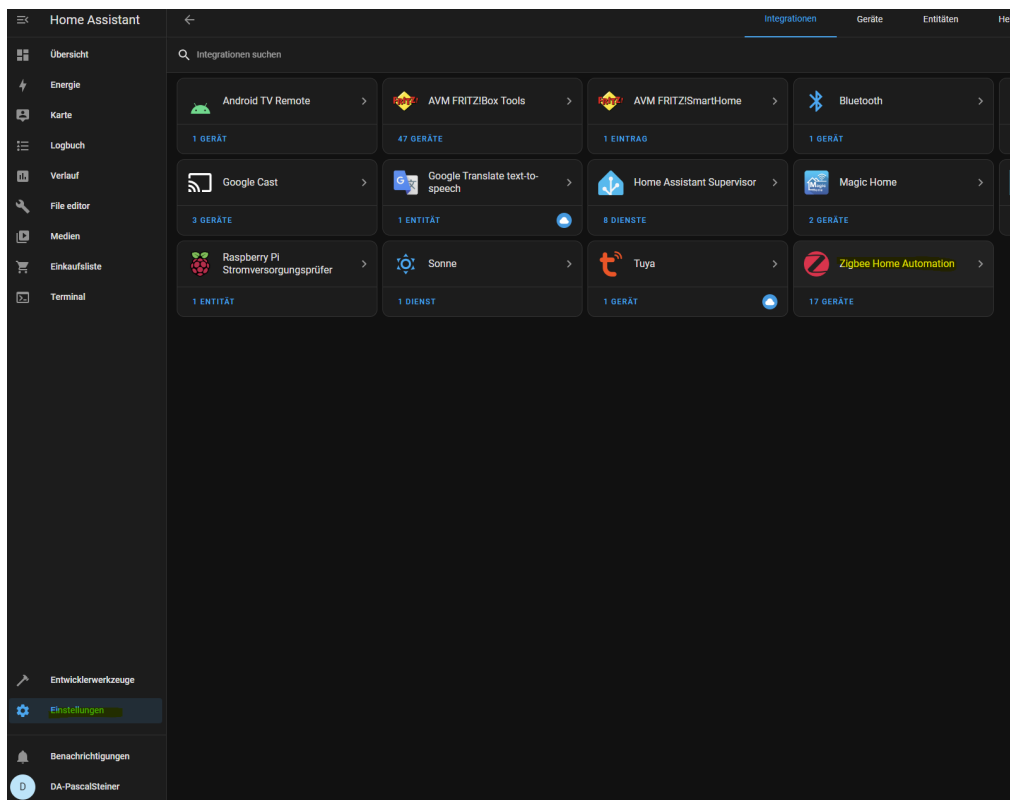


Abbildung 42 Zigbee-Integration

- auf Geräte klicken



Abbildung 43 Zigbee-Integration Konfigurationsübersicht

- danach unten rechts <Gerät hinzufügen> anklicken

Gerät	Hersteller	Modell	Bereich	Integration	Batterie
Bewegungsmelder Büro	_TZ3000_mq44y6z6	TS0202	Büro	Zigbee Home Automation	100 %
Bewegungsmelder Schlafzimmer	_TZ3000_brg9H	TS0202	Schlafzimmer	Zigbee Home Automation	100 %
Entfernenment	_TZ3000_2avhncz	TS011F	Schlafzimmer	Zigbee Home Automation	-
FensterSchlafzimmerrechts	_TZ3000_Augrczll	TS0203	Schlafzimmer	Zigbee Home Automation	100 %
FensterSensor00	_TZ3000_Augrczll	TS0203	Büro	Zigbee Home Automation	100 %
Fernsteuerung	_TZ3290_7y1hkrufopow9tz	TSB1201	Schlafzimmer	Zigbee Home Automation	-
Hauptschalter Lampe	SONOFF	01MINZ8	Schlafzimmer	Zigbee Home Automation	-
HueSwitch	Signify Netherlands B.V.	RWL022	Schlafzimmer	Zigbee Home Automation	100 %
Keller Licht	eWelink	CK-BL782-AL-01(71009_21023.003 1)	Keller Treppe	Zigbee Home Automation	-
Licht	SONOFF	01MINZ8	Büro	Zigbee Home Automation	-
Licht 1	Philips	LWG004	EG Wohnzimmer	Zigbee Home Automation	-
Licht 2	Philips	LWG004	EG Wohnzimmer	Zigbee Home Automation	-
Nachttisch	Philips	LOM006	Schlafzimmer	Zigbee Home Automation	-
Schalter	_TZ3000_Je9mHje	TS0041	Büro	Zigbee Home Automation	100 %
Schalter 1	eWelink	WB01	Schlafzimmer	Zigbee Home Automation	100 %
Temperatur Sensor	eWelink	TH01	Schlafzimmer	Zigbee Home Automation	100 %
Zigbee Coordinator	ZHA	EZSP - Silicon Labs EmberZNet protocol: E16ab5, HL...	Controlroom	Zigbee Home Automation	-

Abbildung 44 Zigbee-Geräte hinzufügen

- in den Pairing-Modus eintreten

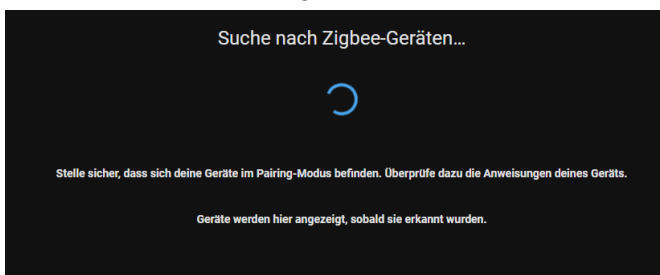


Abbildung 45 Zigbee-Paarungsmodus

- Jetzt sollte ein Zigbee-Gerät eingeschaltet und ggf. in den Paarungsmodus versetzt werden. Es sollte daraufhin erkannt und hinzugefügt werden.

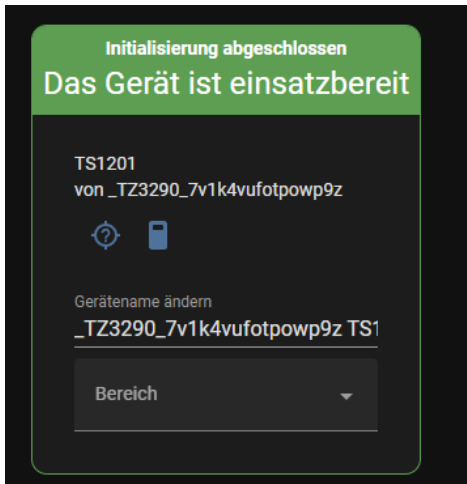


Abbildung 46 ZigBee-Gerät ermittelt

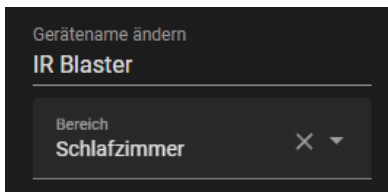


Abbildung 47 Zigbee-Gerät umbenennen und Bereichszuordnung

- Sobald der Rahmen grün erscheint, kann ein Gerätename eingegeben und das Gerät einem Bereich zugeordnet werden.
- Dies sollte für alle Zigbee-fähigen Geräte vorgenommen werden.  
Hinweis: Der Paarungsmodus muss nicht jeweils neu geöffnet werden, denn es können mehrere Geräte auf einmal verbunden werden.

Für die Philips-Hue-Geräte wurde dasselbe Vorgehen angewandt, damit keine Hue Bridge gekauft werden musste, womit weitere Integrationen zu Home Assistant erspart wurden.

### **Tuya-Integration:**

Die Geräte müssen zuerst in der Tuya-Smart- oder der Smart-Life-App hinzugefügt werden.

Es ist ausserdem erforderlich, ein Konto auf der Tuya-IoT(Internet of Things)-Plattform zu erstellen, welches nicht dem Konto in der App entspricht. Dazu wurde folgende Anweisung befolgt:

1. Melde dich auf der Tuya-IoT-Plattform an.
2. Navigiere in der linken Navigationsleiste zu <Cloud> > <Entwicklung>.
3. Klicke auf der erscheinenden Seite auf <Cloud-Projekt erstellen>.
4. Konfiguriere im Dialogfeld <Cloud-Projekt erstellen> Projektname, Beschreibung, Branche und Rechenzentrum. Wähle im Feld <Entwicklungsmethode> <Smart Home> aus der Dropdown-Liste. Im Feld <Rechenzentrum> sollte die Zone selektiert werden, in der sich

der eigene Standort befindet. Nutze die Liste zur Zuordnung von Land/Rechenzentrum, um das passende Rechenzentrum für das eigene Land zu bestimmen.

The screenshot shows a 'Create Cloud Project' dialog box. It contains the following fields:

- \* Project Name:** A text input field with the placeholder 'Enter project name'.
- Description:** A text area with the instruction 'Please briefly describe the functions and scenarios of your project, with no more than 255 words.'
- \* Industry:** A dropdown menu with the placeholder 'Select your industry'.
- \* Development Method:** A dropdown menu with the placeholder 'Select development method'.
- \* Data Center:** A dropdown menu with the placeholder 'Select a data center'.

At the bottom right, there are two buttons: 'Cancel' and 'Create'.

Abbildung 48 Tuya-Cloud-Projekterstellung

5. Klicke auf «Create», um mit der Konfiguration des Projekts fortzufahren.
6. Im Konfigurationsassistenten gilt es sicherzustellen, dass die Dienste «Branchen-Basiser-service», «Smart Home-Basiservice» und «API(Application Programming Interface) zur Benachrichtigung über den Gerätestatus» hinzugefügt sind. Die Liste der APIs sollte wie folgt aussehen:

The screenshot shows the 'Configuration Wizard' at the 'Authorize API Services' step. It includes the following elements:

- Header:** 'Configuration Wizard' and 'Authorize API Services'.
- Text:** 'The platform recommends some API services, and you can remove and select them as needed. The selected unsubscribed API services will be subscribed to and this project will be granted access the API products.'
- Table:** A table with two columns: 'Select API Services' and 'Selected API Service(s) (5)'.
 

Select API Services	Selected API Service(s) (5)
Industry Project Client Service	IoT Core (Free Basic Resource Pack)
Voice Control of Smart Devices (No NLU)	Authorization Token Management
Industry General Organization Management	Industry Basic Service
Body Key Point Recognition	Smart Home Basic Service
Sweeping Robot Open Service	Device Status Notification
Facial Feature Points Recognition	
Face Detection	
- Buttons:** 'Skip' and 'Authorize' at the bottom right.

Abbildung 49 Tuya-API-Konfiguration

7. Klicke auf «Authorize».
8. Wechsle zum Tab «Devices».

## 9. Klicke auf <Link Tuya App Account > Add App Account

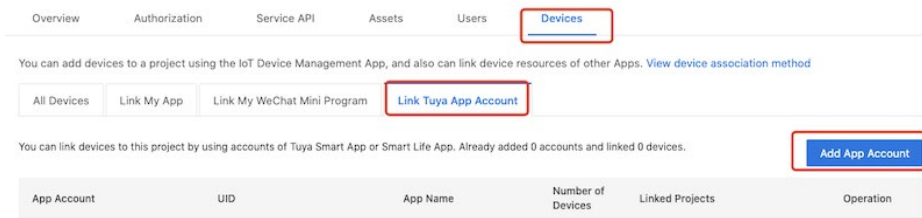


Abbildung 50 Tuya-Account-Link

## 10. Scanne den QR-Code, der in der Tuya-Smart-App oder der Smart-Life-App im Abschnitt <Me> angezeigt wird.

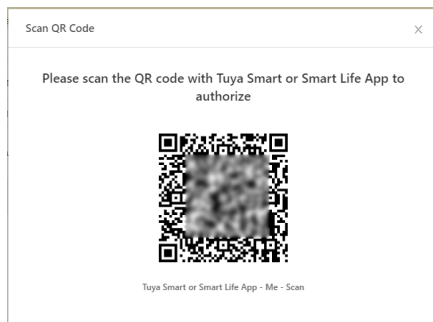


Abbildung 51 Tuya-Link-QR-Code

## 11. Klicke in der App auf <Bestätigen>.

## 12. Um zu überprüfen, ob alles funktioniert hat, wechsele zum Tab <Alle Geräte>. Hier sollten die Geräte aus der App angezeigt werden.

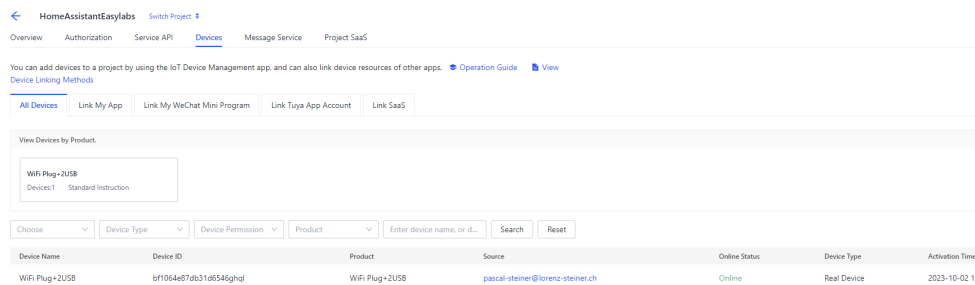


Abbildung 52 Tuya-Device-Übersicht

Klicke auf das erstellte Projekt, um zur Übersichtsseite des Projekts zu gelangen und den Autorisierungsschlüssel zu erhalten. Dieser Schlüssel wird für die Integration in den nächsten Schritten benötigt.

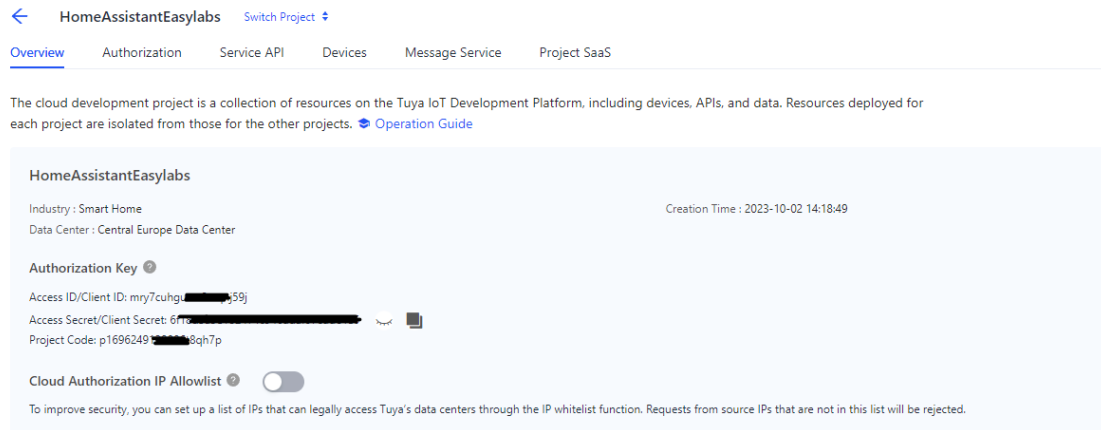


Abbildung 53 Tuya Konfigurationsinformationen

Es gilt nun die Tuya-Integration in Home Assistant zu installieren, wenn dies nicht schon automatisch erfolgt ist durch die automatische Geräteerkennung.

Füge nun einen Hub hinzu. Dazu werden der Autorisierungsschlüssel, die Projekt-ID und das Log-in der Smart-Life-App benötigt.

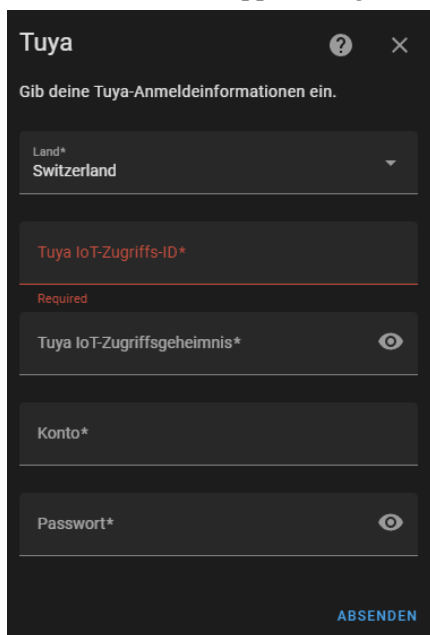


Abbildung 54 HA Tuya Integrationskonfiguration

### **Magic-Home-Integration:**

Diese Integration wurde selbsttätig erkannt und gleich hinzugefügt nach dem Konfigurieren der Fritz!Box in der Basiskonfiguration. Dadurch wurden die WiFi-Controller der Lichter erkannt. Die Integration wurde vorgeschlagen und die erkannten zugehörigen Geräte mussten nur noch benannt werden.

Diese Integration wird verwendet, um die schon installierte Lampe im Schlafzimmer und die LED-Streifen genau zu steuern. Dafür werden weder die Cloud noch Accounts genutzt.

## **7.6 Integration der Sprachassistenten Google Home und Alexa**

Mit Home Assistant Cloud können Nutzer ihre Home-Assistant-Instanz in wenigen Schritten mit Google Assistant und Amazon Alexa verbinden, ohne sich mit dynamischem DNS, SSL-

Zertifikaten oder Portfreigaben auf ihrem Router auseinandersetzen zu müssen. Die Verbindung erfolgt über die Benutzeroberfläche, erfordert jedoch nach einer kostenlosen 30-tägigen Testphase ein kostenpflichtiges Abonnement. Dementsprechend wird vorliegend die manuelle Einrichtung genutzt, um ein weiteres Abonnement zu verhindern.

### 7.6.1 Google Assistant (Nest Mini)

Die Google-Assistent-Integration ermöglicht die Steuerung von Home-Assistant-Geräten über Google Assistant auf Mobilgeräten, Tablets oder Google Home-Geräten.

Um die Google-Assistent-Integration einzurichten, wurden die nachfolgenden Schritte durchgeführt.

#### Google Cloud Plattformkonfiguration (<https://console.actions.google.com>) :

1. Erstelle ein neues Projekt unter den Aktionen in der Google-Konsole.
2. Wähle «Neues Projekt» und vergib einen Namen.
3. Wähle die Option «Smart Home» und klicke auf «Mit dem Erstellen beginnen».
4. Gib der Aktion einen Namen. Home Assistant wird in der Google-Home-App als «[Test] <Aktionsname>» angezeigt.
5. Klicke auf «Übersicht», um zur Übersichtsseite zurückzukehren.
6. Wähle «Ihre Aktion erstellen» und dann «Aktion(en) hinzufügen».
7. Trage deine Home-Assistant-URL ein mit `https://[IHRE HOME ASSISTANT-URL:PORT]/api/google_assistant` im Feld «URL» ([https://ha-easylabs.duckdns.org:8123/api/google\\_assistant](https://ha-easylabs.duckdns.org:8123/api/google_assistant)).
8. Klicke auf «Speichern».
9. Klicke auf die drei Punkte (Auslassungspunkte) in der oberen rechten Ecke und wähle «Projekteinstellungen» aus.
10. Notiere die Projektkennung auf der Registerkarte «Allgemein» der Einstellungsseite.
11. Kehre zur Registerkarte «Übersicht» zurück und wähle «Konto-Verknüpfung einrichten» im Abschnitt «Schnelleinrichtung» aus.
12. Wähle für den Verknüpfungstyp «OAuth und Autorisierungscode» und dann «Weiter» aus.
13. Gib die folgenden Informationen ein:
  - Client-ID: [https://oauth-redirect.googleusercontent.com/r/\[IHRE\\_PROJEKTKENNUNG\]](https://oauth-redirect.googleusercontent.com/r/[IHRE_PROJEKTKENNUNG]) (ersetze «[IHRE\_PROJEKTKENNUNG]» durch die Projekt-ID)
  - Client Secret: Dies ist ein beliebiger Wert, der von Home Assistant nicht benötigt wird.
  - Autorisierungs-URL: `https://[IHRE HOME ASSISTANT-URL:PORT]/auth/authorize` <https://ha-easylabs.duckdns.org:8123/auth/authorize>
  - Token-URL: `https://[IHRE HOME ASSISTANT-URL :PORT]/auth/token` <https://ha-easy-labs.duckdns.org:8123/auth/token>
15. Wähle «Weiter» und dann «Speichern».
16. Wechsle zur Registerkarte «Entwickeln» und wähle oben rechts die Schaltfläche «Test» aus, um die Entwurfsversion der Test-App zu erstellen.

#### Google Cloud Plattformkonfiguration für die Synchronisation (<https://console.cloud.google.com/>) :

1. Gehe zur Google-Cloud-Plattform.
2. Wähle «Projekt auswählen».

3. Wähle in Schritt eins das erstellte Projekt aus.
4. Navigiere zum Menü «APIs and Services» und dann zu «Credentials».
5. In der Ansicht «Credentials» wähle «Create credentials» und dann «Service account».
6. Vergib einen Namen für den Service-Account.
7. Wähle «Create» und «Continue».
8. Wähle die Rollen «Service Accounts» und «Service Account Token Creator».
9. Wähle «Continue».
10. Wähle «Done».
11. Unter «Service Accounts» sollte nun ein Konto namens «[Servicename]@[Projektname].iam.gserviceaccount.com» angezeigt werden.
12. Klicke auf den Stift zum Bearbeiten des Service-Accounts.
13. Gehe zu «Keys» und füge einen Schlüssel hinzu.
14. Erstelle einen privaten Schlüssel im JSON-Format.
15. Dieser wird als JSON-Datei heruntergeladen.
16. Benenne die Datei in «SERVICE\_ACCOUNT.JSON» um.
17. Füge diese Datei in den Konfigurationsordner ein (dieser Ordner entspricht der configuration.yaml-Datei in Home Assistant).

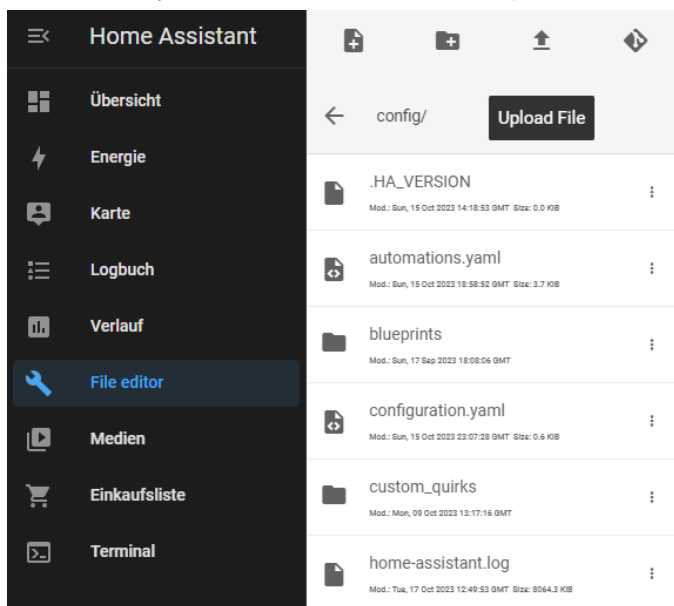


Abbildung 55 Upload des Account-Files

18. Gehe zurück zur Google-Cloud-Plattform und wählen «Schliessen».
19. Wähle «Speichern».
20. Gehe zur «Google API Console».
21. Wähle unter «Aktionen» in der Google-Konsole das erstellte Projekt aus.
22. Aktiviere die «HomeGraph API».

### Konfiguration von Home Assistant:

Fügen google\_assistant-Integrationskonfiguration zur configuration.yaml-Datei hinzu und starte Home Assistant neu.

**Dazu gilt es folgenden Block in configurations.yaml hinzuzufügen:**

google\_assistant :

```
project_id : [PROJEKT- ID] # meine ist «Home Assistant-10218»  
service_account : !include SERVICE_ACCOUNT.JSON  
report_state: true # aktiviert die Synchronisation  
exposed_domains:  
- switch  
- light
```

Unter «Exposed Domains» können Entitätentypen definiert werden, welche zu Google synchronisiert werden. Es können auch Szenen und Routinen angegeben werden, was ich später gemacht habe, damit ich gewisse Scherzroutinen auslösen kann. Dies musste manuell in der Konfiguration eingefügt werden, weil Home Assistant bei der Integration für Sprachassistenten die eigene Cloud promotet und keine Konfiguration mit GUI für die manuelle Einrichtung bietet. Nach dem Neustart war die Integration jedoch aktiv und die Geräte konnten synchronisiert werden.

#### **Dienste konnten nach folgendem Schema in der Google Home-App hinzugefügt werden:**

1. Öffne die Google-Home-App.
2. Wähle oben links die Schaltfläche «+» und dann «Gerät einrichten» aus.
3. Wähle auf der Seite «Gerät einrichten» die Option «Mit Google funktioniert» aus. Dort sollte [Test] «Aktionsname» aufgeführt sein. Die Auswahl leitet den Nutzer zu einem Browser weiter, wo er sich bei seiner Home-Assistent-Instanz anmelden kann. Von hier gelangt er zu einem Bildschirm zurück, auf dem Räume und Spitznamen für die eigenen Geräte festgelegt werden können, wenn dies gewünscht ist.

Sollte Home Assistant bereits auf dem Smartphone installiert sein, muss die App im Vorfeld deinstalliert werden, da die Anmeldung von Google zu Home Assistant sonst die App öffnet und nicht mehr korrekt zurückleitet.

### **7.6.2 Amazon Alexa**

Es gibt verschiedene Möglichkeiten, wie Amazon Alexa und Home Assistant in Kombination genutzt werden können.

- Erstelle ein Alexa Smart Home Skill, um Lichter usw. zu steuern.
  - Unterstützung von Äusserungen ohne den Skill-Namen
  - Unterstützung der Steuerung von Geräten über Alexa-Routinen
  - Unterstützung beim Anzeigen und beim Steuern von Geräten über die Alexa-Mobile-App
- Erstelle ein benutzerdefiniertes Alexa Skill, um benutzerdefinierte Befehle einzurichten.
- Erstelle eine neue Quelle für das Flash-Briefing.
- Alternative: Verwende die Emulated-Hue-Integration, um Alexa vorzutäuschen, dass Home Assistant ein Philips-Hue-Hub ist.

Im Projekt wurde die erste Variante implementiert, da diese die besten Konfigurationsmöglichkeiten bietet.

Die manuelle Einrichtung der Integration mit Amazon Alexa erfordert bei allen Varianten einige Voraussetzungen:

- Es ist ein Amazon-Developer-Account nötig.
- Für das Erstellen benutzerdefinierter Befehle und eines Flash-Briefing muss die Home-Assistent-Instanz über HTTPS aus dem Internet erreichbar sein.

- Ein AWS(Amazon Web Service)-Konto wird benötigt, wenn die Smart-Home-Skill-API verwendet werden soll. Ein Teil der Smart-Home-Skills wird auf AWS Lambda gehostet. Letzterer Dienst ist bis zum Erreichen von einer Million Anfragen und 1 GB an ausgehendem Datentransfer pro Monat kostenlos.

Obwohl der Account gratis bleibt, muss eine Kreditkarte hinterlegt werden. Karten, die mit 3D Secure gesichert sind, funktionieren hier nicht, was im Projekt eine Herausforderung darstellte. Die Installation ist kompliziert, deshalb bin ich strikt nach der Anleitung unter [https://www.home-assistant.io/integrations/alexa.smart\\_home/](https://www.home-assistant.io/integrations/alexa.smart_home/) vorgegangen. Nachfolgend werden nur die wesentlichen Punkte und Konfiguration in Home Assistant zusammengefasst.

### **Erstellung eines Skills bei Amazon Alexa Smart Home**

1. Melde dich in der Alexa Developer Console mit dem Amazon-Konto an, das du auch auf Alexa-Geräten und in der App verwendest.
2. Erstelle einen neuen Skill mit dem Skilltyp <Smart Home>.
3. Wähle den Payload-Typ <v3> und notiere dir die Skill-ID.
4. Erstelle eine AWS-Lambda-Funktion, die Anfragen von Alexa an Home Assistant weiterleitet. Stelle sicher, dass eine AWS-Rolle für Lambda erstellt wird und die erforderlichen Berechtigungen zugewiesen sind.
5. Implementiere den Python-Code für die Lambda-Funktion, der die Anfragen von Alexa verarbeitet und die Antwort an die Smart-Home-Funktion in Home Assistant weiterleitet.
6. Konfiguriere Umgebungsvariablen für die Lambda-Funktion – einschliesslich der Basis-URL für die Home-Assistant-Instanz, einer Einstellung zur Überprüfung von SSL-Zertifikaten und Debug-Optionen.
7. Teste die Lambda-Funktion, um sicherzustellen, dass sie erfolgreich auf Anfragen von Alexa reagiert.

### **Konfigurieren des Smart-Home-Service-Endpunkts**

1. Kopiere die ARN(Amazon Resource Name) der Lambda-Funktion und gib sie in den Smart-Home-Service-Endpunkt der Alexa Developer Console ein.
2. Stelle sicher, dass das Home-Assistant-System über HTTPS erreichbar ist.

### **Kontenverknüpfung**

1. Richte die Kontenverknüpfung ein, um das eigene Amazon-Konto mit dem Home Assistant-Konto zu verknüpfen.
2. Verwende den Oauth-2.0-Standard, um die Autorisierungs-URI(Uniform Resource Identifier) und die Zugriffs-Token-URI für die Kontenverknüpfung festzulegen.
3. Stelle sicher, dass das SSL-Zertifikat gültig und vertrauenswürdig ist, da selbstsignierte Zertifikate nicht funktionieren.
4. Aktiviere die Kontenverknüpfung über das Aktivieren der Smart-Home-Fähigkeit in der Alexa-App.

### **Testen und Verwenden**

1. Teste, ob Alexa die eigenen Home-Assistant-Geräte erkennt. Dafür gilt es Alexa dazu aufzufordern, Geräte zu erkennen.
2. Verwende die Alexa-App oder ein Echo-Gerät, um die Smart-Home-Geräte über Sprachbefehle zu steuern, z. B. mit <Alexa, schalte das Schlafzimmerlicht ein.>

## Zusätzliche Überlegungen

- Es ist empfehlenswert, die Konfiguration in Home Assistant zu überprüfen und hier zu steuern, welche Geräte sowie Entitäten für Alexa freigegeben werden.
- Für Entwickler sind Details zur proaktiven Modus-Unterstützung und mögliche Anpassungen des Python-Codes für die Lambda-Funktion verfügbar.

Diese Schritte ermöglichen es, Amazon Alexa nahtlos in die Smart-Home-Umgebung zu integrieren und Geräte sowie Automatisierungen über Sprachbefehle zu steuern. Dabei gilt es die Anforderungen und die Sicherheitsempfehlungen für eine reibungslose Integration zu beachten.

## 7.7 Erstellung von Automatisierungen

Die Erstellung der Automationen habe ich immer mit dem Automation Editor begonnen, da dieser übersichtlich und intuitiv gestaltet ist. Bei Bedarf sind zahlreiche Anleitungsvideos im Internet auffindbar.

Bei Home Assistant stehen zwei Vorlagen für die Automatisierung zur Verfügung, zum Beispiel für einen Bewegungsmelder. Diese Vorlage wurde zu Beginn von mir genutzt, jedoch ist das Ergebnis einfach; es erfolgen keine Checks. Somit würde ich in der Nacht immer das Licht einschalten, wenn ich mich im Bett bewege. Deswegen habe ich die Automatisierung angepasst mit Ruhezeiten von 22:00 Uhr bis 06:00, zu denen der Bewegungsmelder inaktiv ist.

Nach weiterer Recherche bin ich auf Blueprints (Vorlagen) der Community gestossen. Aus dieser habe ich bisher zwei Blueprints importiert. Der erste löst die Probleme des Bewegungsmelders, da mit dieser Vorlage unter anderem Ruhezeiten definiert sowie zusätzliche Sensoren über den Lichtwert abgefragt und konfiguriert werden können. Da ich bemerkt habe, dass meine Bewegungsmelder keinen Lux (Lichtwert) messen können, habe ich sie anhand der Sonnenintegration aktivieren und deaktivieren lassen. So wird das Licht am Tag nicht eingeschaltet. Die andere verwendete Vorlage ist für den Hue Dimmer Switch bestimmt, da dieser wegen der Zigbee-Integration nicht alle Knöpfe als direkte Entität freigibt.

Die restlichen Automationen wurden nicht via GUI erstellt, sondern direkt in der `automations.yaml`. Alle Automationen sind im Anhang 3 angefügt, in Abbildung 56 folgt aber eine grobe Übersicht.

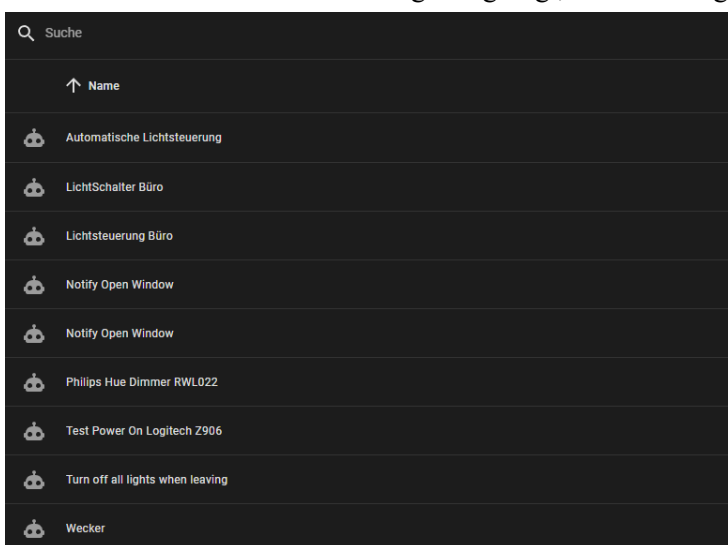


Abbildung 56 Übersicht der Automatisierungen

Zum Verständnis wie so eine Automation direkt mittels YAML Code erstellt werden kann, wird nun anhand der Automatisierung für das Alarmieren des öffnen eines Fensters aufgezeigt. Dazu am besten bei einem der der betroffenen Sensoren auf Automatisierungen gehen.

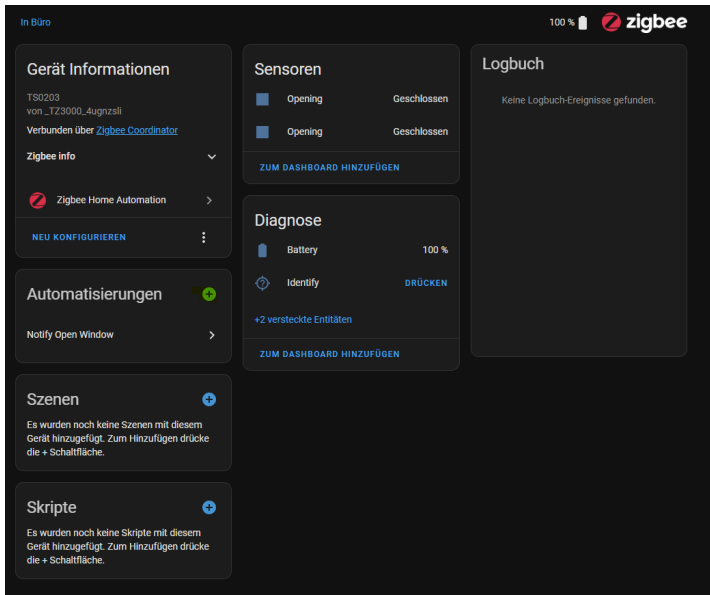


Abbildung 57 Erstellung neuer Automation

Im anschliessenden Popup kann irgendetwas ausgewählt werden da dies sowieso überschrieben wird. Im nun offenen Visuellen Editor oben rechts auf die drei Punkte klicken und Als YAML bearbeiten auswählen

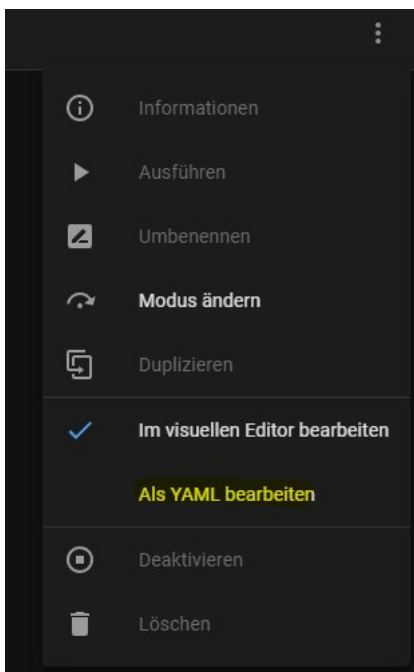


Abbildung 58 Automationsansicht zum YAML Editor wechseln

Anschliessend kann nun die gewünschte Routine erstellt werden. Der Aufbau der Automation YAML ist immer gleich:

```
alias: [NAME Für Automatisierung]
description: «BESCHREIBUNG»
```

```

trigger: [] (Ein oder mehrer Auslöser)
condition: [] (Keine oder mehrere Bedingungen)
action: [] (Ausführende Aktionen)
mode: [GEWÜNSCHTER Modus]
In unserem Beispiel sieht der Code so aus:
alias: Einbruchswarung
description: ""
trigger:
  - type: opened
    platform: device
    device_id: cb696c613090c1adca68a8ae8c3cc1b9
    entity_id: 3dc914e0441a6af4dccbe133d0eec50e
    domain: binary_sensor
  - type: opened
    platform: device
    device_id: b973f53fc46a0fdf48f5261b3e0cbae1
    entity_id: e21c51af8246ca0622ce27e846237e6f
    domain: binary_sensor
condition:
  - condition: time
    after: "23:00:00"
    before: "05:30:00"
action:
  - service: notify.notify
    data:
      message: Fenster wurden geöffnet möglicher Einbruch
      title: Einbruchswanung
mode: single

```

Wichtig zu beachten ist bei YAML die richtige Einrückung (Nächst tiefere Ebenen mit zwei Spaces)

Wie die Device\_id oder welchen Service man verwendet, kann man bei den Geräten und Entitäten herausfinden. Es ist aber anzuraten hierzu die Dokumentation von Home Assistant zu nutzen.

Anschliessen Speichern und Testen

Hinweis: In der automation.yaml Datei wird noch eine eindeutige ID zu der Automation erstellt

## 7.8 Dashboard konfigurieren

Für diesen Schritt habe ich folgende Anleitung befolgt:

Öffne die Home Assistant-Oberfläche im Webbrowser oder in der App.

### **Dashboards verwalten:**

Klicke oben links auf das Menüsymbol (drei horizontale Linien) und wähle ‹Übersicht› aus. Dies ist das Standard-Dashboard.

Um weitere Dashboards hinzuzufügen, klicke auf **«Konfiguration»** und dann auf **«Dashboards»**. Hier können neue Dashboards erstellt und bestehende bearbeitet werden.

**Bearbeiten des Dashboards:**

Um das Dashboard zu bearbeiten, klicke auf das Stiftsymbol oder oben rechts bei den drei Punkten auf **«Dashboard bearbeiten»**. Nun können Karten und Ansichten hinzugefügt oder vorhandene Elemente angepasst werden.

**Hinzufügen von Karten:**

Klicke auf das **«+»**-Symbol oder auf **«Karte hinzufügen»**. Wähle die gewünschte Kartenart aus, z. B. **«Entitäten»**, **«Grafiken»** oder **«Kamera»**. Gib die erforderlichen Informationen ein, um die Karte zu konfigurieren – z. B. die Entität oder das Gerät, das dargestellt werden soll.

**Erstellen von Ansichten:**

Um zusätzliche Ansichten zu erstellen, klicke auf das **«+»**-Symbol oberhalb der Karten. Gib den Ansichtsnamen ein und lege fest, welche Karten in dieser Ansicht angezeigt werden sollen.

**Anordnen von Karten und Ansichten:**

Die Kartenreihenfolge und -nummer können geändert werden, um die Anordnung anzupassen. Dies ermöglicht es, das Dashboard nach individuellen Wünschen zu gestalten. Die Ansichten können mit den Pfeilen nach links und rechts verschoben werden, um die gewünschte Sortierung zu erhalten.

**Speichern und Übernehmen:**

Speichere die Änderungen ab und klicke auf **«Fertig»**, um das aktualisierte Dashboard zu sichern. Ich habe beim Standard-Dashboard **«Übersicht»** fünf Ansichten erstellt. Die erste bildet eine allgemeine Gesamtübersicht mit den wesentlichen Schaltern und relevanten Informationen.

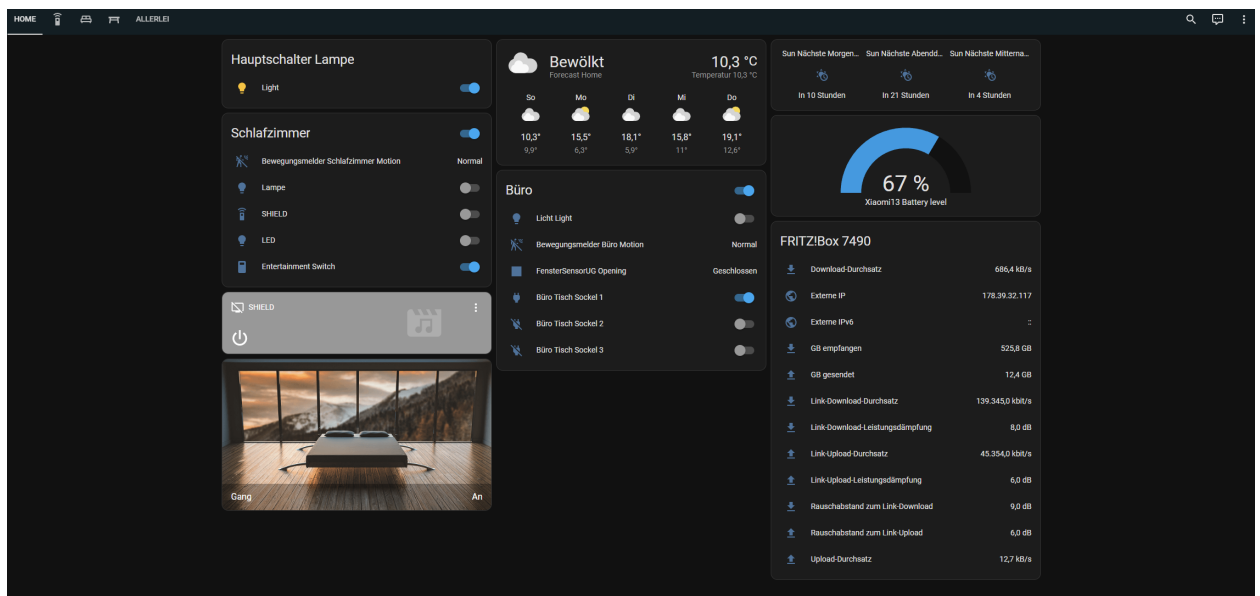


Abbildung 59 Standard-Dashboard

Anknüpfend habe ich ein eigenes Dashboard nur für die Android-TV-Fernbedienung erstellen lassen. Dies gelang durch die Integration von Android TV, indem ich dort auf «zum Dashboard hinzufügen» geklickt habe.

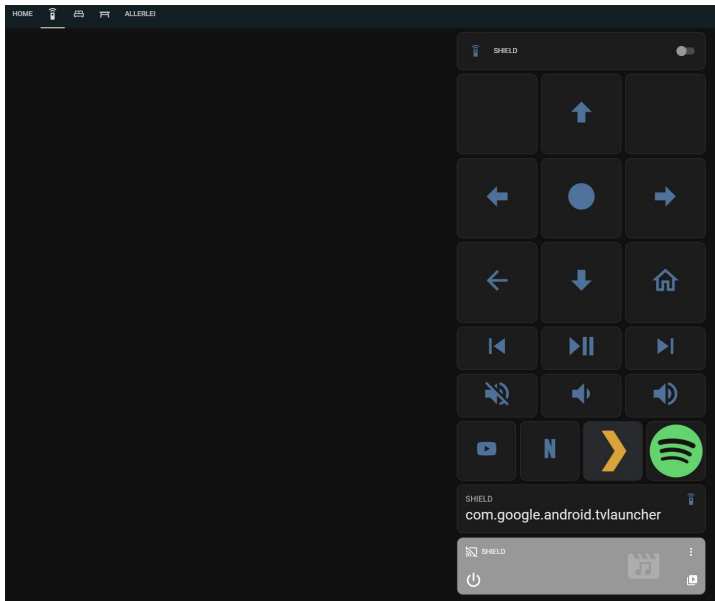


Abbildung 60 Remote-Dashboard

Im Allgemeinen konnten einzelne Entitäten bei den Geräten direkt zum Dashboard hinzugefügt werden.

Anschliessend habe ich je eine Ansicht meiner beiden Räume und ein allgemeines Dashboard erstellt, welches ich zum Testen und für weitere Informationen nutze.

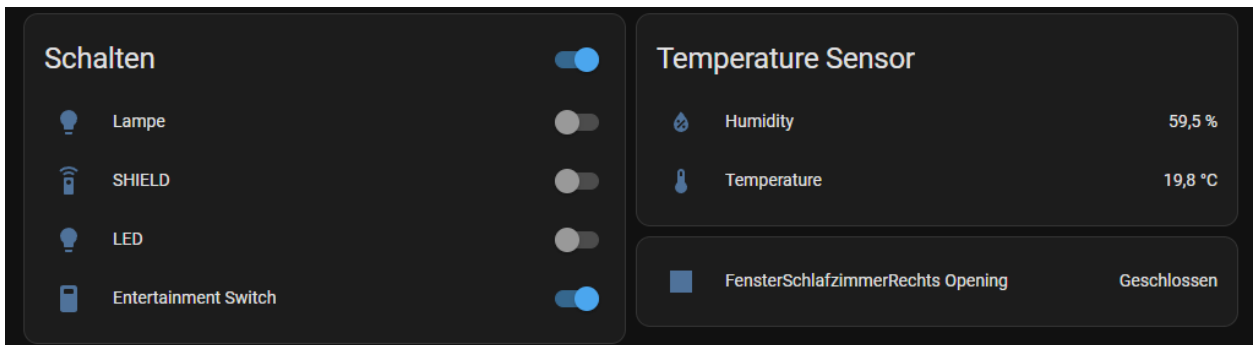


Abbildung 61 Schlafzimmer-Dashboard

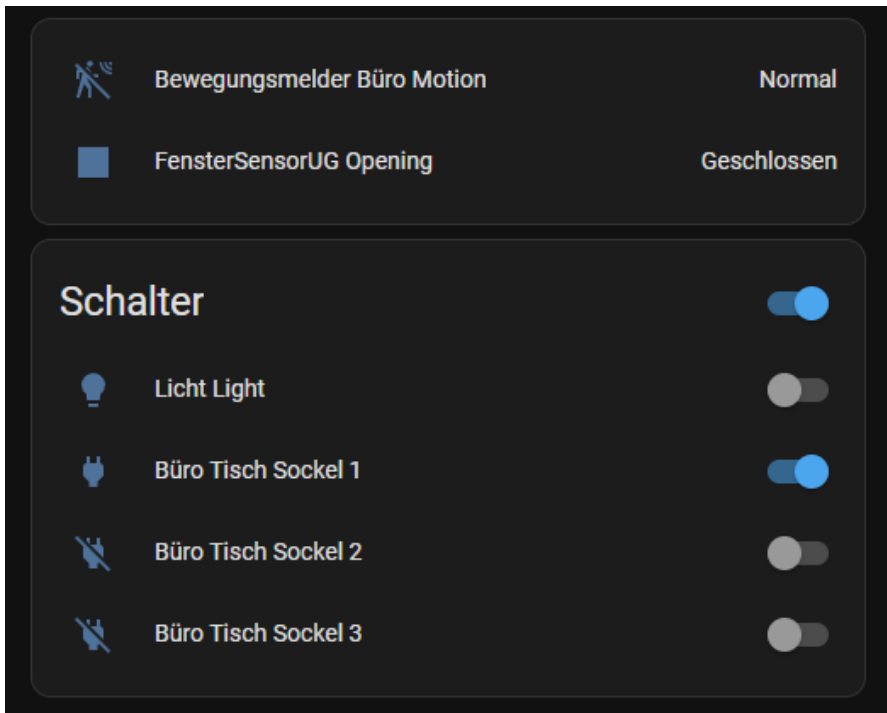


Abbildung 62 Büro-Dashboard

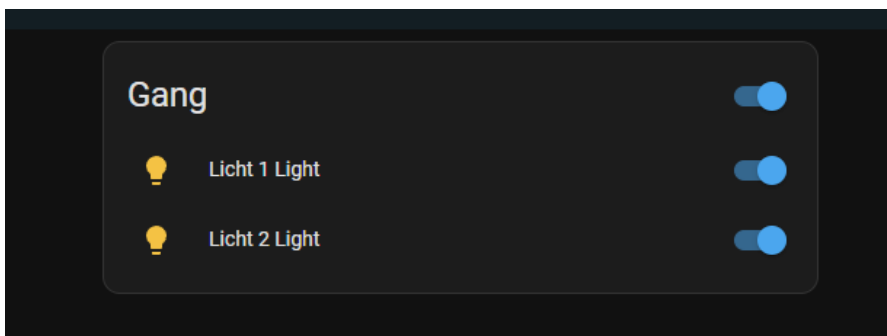


Abbildung 63 Allerlei-Dashboard mit Testkarte

Das Energie-Dashboard, welches auch standardmässig vorhanden ist, habe ich im Assistant durchgeführt, jedoch senden meine Steckdosentypen keine Daten zum aktuellen Verbrauch/zur Last. Nach Prüfung der Gerätetypen ist bekannt, dass diese Smart Plugs die entsprechende Funktion nicht aufweisen. Als Lösung könnte der statische Energieverbrauch gezahlt und über Hard Coding mit den Steckern verknüpft werden, was ich aber nicht gemacht habe. Für den Ausbau müssten entweder alle Smart-Stecker fähig sein, die Daten auszulassen, oder es sollte direkt der Stromzähler ausgemessen werden. Alternativ kann mittels eines Sensors, welcher auf die Stromphasen gesteckt wird, der summierte Stromverbrauch erfasst werden.

Als Alternative für die Statistik habe ich den Inverter der Solaranlage ausgelesen, weshalb ich nur diesen anzeige. Jedoch stimmen die Daten noch nicht komplett: Da zwei Strings (Zuleitungen der Solarzellen) genutzt werden, gibt es mehrere Werte, welche Solarmax normalerweise addiert. Mit meinem gebauten Sensor, welcher ein TCP-Payload versendet, erhalte ich aktuell nur Daten von einem String. Bisher habe ich noch keine bessere Lösung gefunden, alle Daten zu erhalten, da im Manual von Solarmax 10 mt2 nur beschränkt TCP-Pakete definiert sind und ich mit diesen bisher

nur die Daten der ersten Zuleitung beziehen konnte. Konfiguration dazu sind im Anhang 3 einsehbar.

Für die Erstellung des TCP-Sensors musste ich das Protokoll verstehen, was ich durch einen Community-Post im Home-Assistant-Forum gelernt habe. Die Links dazu sind als Quellen angegeben.

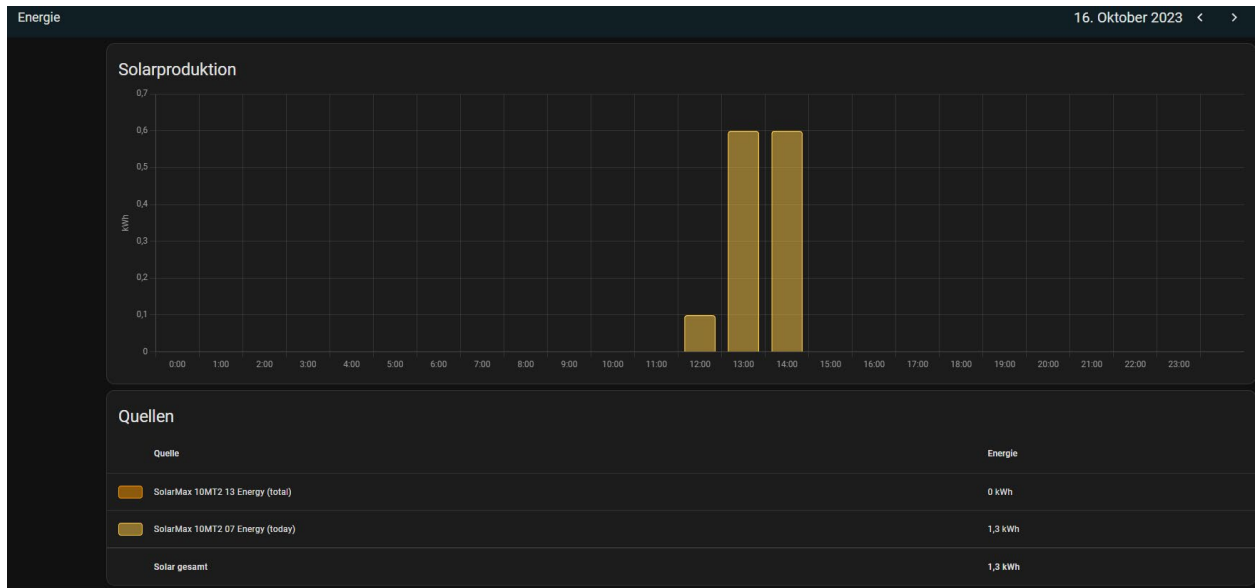


Abbildung 64 Energie-Übersicht

## 7.9 Zusammenfassung der Umsetzung

Da die vorangehenden Anleitungen nicht alle Arbeitsschritte enthalten, fasse ich nachfolgend den gesamten Prozess zusammen.

Zunächst hat mir die Dokumentation geholfen, jedoch hatte ich erstmalig viel ausprobiert und die Dokumentationen von Home Assistant später zur Hand genommen. Das führte dazu, dass ich einige Schritte doppelt ausgeführt habe, damit sie korrekt umgesetzt wurden. Das Einrichten und das Integrieren waren leicht nachzubauen. Schwierigkeiten entstanden beim Automatisieren und beim Erstellen von Abläufen, was viel Zeit und zahlreiche Tests erforderte.

Zudem war der Infrarotsensor/-Aktor schwieriger zu bedienen als gedacht. Nach über zwölf Stunden des Ausprobierens, des Recherchierens und des Programmierens habe ich seine Einrichtung aufgegeben. Ich wollte den Sensor ursprünglich einsetzen, um meine Musikanlage zu steuern. Dieses Ziel habe ich nicht erreicht.

Des Weiteren hat mir Amazon Alexa Mühe bereitet, aber die meisten diesbezüglichen Herausforderungen waren auf ein Kreditkartenproblem zurückzuführen.

Ein weiteres Problem bildete der Temperatursensor, welcher kaputt angekommen war. Diesen musste ich reparieren – der Knopf für Resetting sowie Pairing war halb abgelöst und musste neu verlötet werden.

## 8 Projektabschluss

Es ist schwierig, schriftlich abzubilden, wie das Endprodukt dieses Projekts aussieht und funktioniert. Deshalb wird nachfolgend das gesamte Projekt bewertet. Es wird unter Einsatz von Testfällen geprüft, ob die Zielsetzung erreicht wurde. Ferner erfolgen Reflexionen und ein Ausblick.

### 8.1 Projektüberwachung

Aufgrund meiner privaten Situation, in der ich im gleichen Zeitraum auch andere Projekte bearbeiten musste, hat sich der Ablauf um fast eine Woche verschoben. Einige der hindernden Termine hatte ich bei der Planung berücksichtigt, andere jedoch nicht oder ich habe sie falsch eingeschätzt. Ich habe bei zwei Events mitgeholfen, welche jeweils vier Tage lang dauerten, wobei ich bei der zweiten Veranstaltung einen Tag früher ging, um an meinem Projekt zu schreiben. Zusätzlich hatte ich eine Woche Pikettdienst bei der Arbeit – mit mehreren Einsätzen unter der Woche und einem grossen Einsatz am Wochenende. Der Pikettdienst wurde in der Planung vergessen, hatte allerdings deutliche Auswirkungen.

Zudem wurden einzelne Punkte der Arbeit selbst unterschätzt, denn sie haben mehr Zeit in Anspruch genommen als gedacht. So habe ich den Zeitaufwand für das Dokumentieren unterschätzt. Dennoch konnte der Fortschritt stetig mit dem Projektablaufplan abgeglichen werden. Trotz der Verzögerungen wurden für das Projekt vorgehend zeitliche Reserven eingeplant, welche gebraucht wurden.

Es erwies sich als ineffizient, dass ich nicht von Anfang an priorisiert habe, sondern erst nach dem ersten Expertengespräch. Zukünftig werde ich vorab alle Eckdaten definieren und eine Priorisierung erstellen sowie gegebenenfalls weitere Informationen für die Planung beschaffen. Wie gehabt werde ich Reserven einplanen. Dokumentation werde ich doppelt so viel Zeit zuordnen wie ursprünglich.

Zudem werde ich darauf achten, dass privat weniger Projekte gleichzeitig bearbeitet werden müssen, indem ich Termine verschiebe.

### 8.2 Evaluation der Zielerreichung

Nachfolgend wurden Test-Cases anhand der Zielvorgaben definiert und anschliessend in einer Übersicht zusammengefasst sowie bewertet. Für die Erzeugung der Testfälle wurde ChatGPT verwendet unter Eingabe der Zieldefinition. Es wurden über 50 Testfälle generiert. Diese habe ich auf 23 Stück reduziert habe, da einige Tests fast auf dasselbe ausgerichtet und somit redundant waren.

Die KI(Künstliche Intelligenz) habe ich mit Absicht genutzt, um die Testfälle nicht auf die angestrebte Funktionalität auszurichten, sondern unvoreingenommen prüfen zu können, ob alle Ziele erreicht wurden.

#### 8.2.1 Definition der Testfälle

##### Testfall Nr. 1

- Bezeichnung: Installation und Konfiguration
- Beschreibung: Überprüfe, ob die Installation und die Konfiguration des Smart-Home-Systems erfolgreich durchgeführt wurden.
- Ziel: Installation des Smart-Home-Systems
- Testschritte:
  1. Installiere das Smart-Home-System gemäss den Anweisungen.

2. Konfiguriere die Systemeinstellungen.

- Erfüllungskriterium: Die Installation und die Konfiguration wurden erfolgreich abgeschlossen.

**Testfall Nr. 2**

- Bezeichnung: Integration von Zigbee-Geräten
- Beschreibung: Überprüfe, ob mindestens drei verschiedene Zigbee-fähige Geräte verschiedener Hersteller erfolgreich integriert wurden.
- Ziel: Integration von Zigbee-Geräten
- Testschritte:
  1. Integriere mindestens drei verschiedene Zigbee-fähige Geräte.
  2. Stelle sicher, dass die Geräte ordnungsgemäss funktionieren.
- Erfüllungskriterium: Mindestens drei verschiedene Zigbee-Geräte sind erfolgreich integriert und funktionieren.

**Testfall Nr. 3**

- Bezeichnung: Sprachsteuerung
- Beschreibung: Überprüfe, ob die Sprachsteuerung über Google Assistant und Alexa erfolgreich implementiert wurde.
- Ziel: Implementierung der Sprachsteuerung
- Testschritte:
  1. Aktiviere die Sprachsteuerung über Google Assistant und Alexa
  2. Führe Sprachbefehle aus und überprüfe die Reaktion des Systems.
- Erfüllungskriterium: Die Sprachsteuerung ist erfolgreich aktiviert und funktioniert.

**Testfall Nr. 4**

- Bezeichnung: Beleuchtungssteuerung
- Beschreibung: Überprüfe, ob die Beleuchtungssteuerung einschliesslich der Fernsteuerung erfolgreich implementiert wurde.
- Ziel: Realisierung der Beleuchtungssteuerung
- Testschritte:
  1. Steuern die Beleuchtung manuell.
  2. Teste die Fernsteuerung der Beleuchtung.
- Erfüllungskriterium: Die Beleuchtungssteuerung ist erfolgreich implementiert.

**Testfall Nr. 5**

- Bezeichnung: Temperaturüberwachung
- Beschreibung: Überprüfe, ob die Temperaturüberwachung erfolgreich implementiert wurde.
- Ziel: Realisierung der Temperaturüberwachung.
- Testschritte:
  1. Überwache die Raumtemperatur.
  2. Überprüfe, ob die Werte mit denen auf einem analogen Thermometer übereinstimmen.
- Erfüllungskriterium: Die Temperaturüberwachung ist erfolgreich implementiert.

**Testfall Nr. 6**

- Bezeichnung: Bedienung von Geräten
- Beschreibung: Überprüfe, ob die Bedienung von Geräten erfolgreich implementiert wurde.
- Ziel: Realisierung der Bedienung von Geräten
- Testschritte:
  1. Bediene verschiedene Geräte im System.
  2. Überprüfe die Zuverlässigkeit der Gerätesteuerung.
- Erfüllungskriterium: Die Bedienung von Geräten ist erfolgreich implementiert und zuverlässig.

**Testfall Nr. 7**

- Bezeichnung: automatische Lichtsteuerung
- Beschreibung: Überprüfe, ob die automatische Lichtsteuerung basierend auf Anwesenheit und Bewegung im Raum erfolgreich funktioniert.
- Ziel: Realisierung der automatischen Lichtsteuerung
- Testschritte:
  1. Überprüfe die automatische Ein- und Ausschaltung von Licht basierend auf Anwesenheit und Bewegung im Raum.
- Erfüllungskriterium: Die automatische Lichtsteuerung ist erfolgreich implementiert und funktioniert.

**Testfall Nr. 8**

- Bezeichnung: zusätzliche smarte Schalter
- Beschreibung: Überprüfe, ob zusätzliche smarte Schalter für weitere Geräte und Funktionen verfügbar sind.
- Ziel: Realisierung von zusätzlichen smarten Schaltern
- Testschritte:
  1. Überprüfe die Verfügbarkeit und die Funktionsfähigkeit von zusätzlichen smarten Schaltern.
- Erfüllungskriterium: Zusätzliche smarte Schalter sind erfolgreich implementiert und funktionieren.

**Testfall Nr. 9**

- Bezeichnung: sprachgesteuerte Schaltungen
- Beschreibung: Überprüfe, ob Geräte und Funktionen mit Sprachbefehlen gesteuert werden können.
- Ziel: Realisierung von sprachgesteuerten Schaltungen
- Testschritte:
  1. Führe Sprachbefehle zur Steuerung von Geräten und Funktionen durch.
- Erfüllungskriterium: Die sprachgesteuerten Schaltungen sind erfolgreich implementiert und funktionieren.

**Testfall Nr. 10**

- Bezeichnung: Datensammlung
- Beschreibung: Überprüfe, ob Daten über den Stromverbrauch, die Temperatur und die Bewegungen im Haus erfolgreich gesammelt werden.
- Ziel: Realisierung der Datensammlung

- Testschritte:
  1. Überprüfe die Erfassung von Daten über den Stromverbrauch.
  2. Überprüfe die Erfassung von Daten zur Raumtemperatur.
  3. Überprüfe die Erfassung von Bewegungsdaten im Haus.
- Erfüllungskriterium: Die Datensammlung ist erfolgreich implementiert und funktioniert.

**Testfall Nr. 11**

- Bezeichnung: Sicherheit
- Beschreibung: Überprüfe, ob das Smart-Home-System das Haus vor Eindringlingen und anderen Gefahren schützt.
- Ziel: Realisierung der Sicherheitsfunktionen
- Testschritte:
  1. Aktiviere die Sicherheitsfunktionen.
  2. Teste die Reaktion auf Sicherheitsereignisse.
- Erfüllungskriterium: Die Sicherheitsfunktionen sind erfolgreich implementiert und schützen das Haus.

**Testfall Nr. 12**

- Bezeichnung: spielerische Automatismen
- Beschreibung: Überprüfe, ob spielerische Automatismen verfügbar sind, die das Leben im Haus angenehmer machen.
- Ziel: Realisierung von spielerischen Automatismen
- Testschritte:
  1. Überprüfe die Verfügbarkeit von spielerischen Automatismen.
  2. Teste die Funktionalität und die Unterhaltungswirkung.
- Erfüllungskriterium: Spielerische Automatismen sind erfolgreich implementiert und unterhaltsam.

**Testfall Nr. 13**

- Bezeichnung: grafische Übersicht
- Beschreibung: Überprüfe, ob eine grafische Übersicht aller Schalter, Sensoren, Automatismen und Datensammlungen verfügbar ist.
- Ziel: Realisierung einer grafischen Übersicht
- Testschritte:
  1. Überprüfe die Verfügbarkeit der grafischen Übersicht.
  2. Teste die Benutzerfreundlichkeit und die Informationsdarstellung.
- Erfüllungskriterium: Die grafische Übersicht ist erfolgreich implementiert und benutzerfreundlich.

**Testfall Nr. 14**

- Bezeichnung: Marktanalyse
- Beschreibung: Überprüfe, ob eine Marktanalyse über DIY-smart-Home-Geräte erfolgreich durchgeführt wurde.
- Ziel: Durchführung der Marktanalyse
- Testschritte:
  1. Analysiere den Markt für DIY-smart-Home-Geräte.

## 2. Dokumentiere die Ergebnisse der Marktanalyse.

- Erfüllungskriterium: Die Marktanalyse wurde erfolgreich durchgeführt und dokumentiert.

### **Testfall Nr. 15**

- Bezeichnung: Untersuchung von Sensor- und Aktortypen
- Beschreibung: Überprüfe, ob verschiedene Sensor- und Aktortypen untersucht wurden sowie eine Vergleichsmatrix mit Vor- und Nachteilen erstellt wurde.
- Ziel: Untersuchung von Sensor- und Aktortypen
- Testschritte:
  1. Untersuche verschiedene Sensor- und Aktortypen.
  2. Erstelle eine Vergleichsmatrix mit Vor- und Nachteilen.
- Erfüllungskriterium: Die Untersuchung wurde erfolgreich durchgeführt und die Vergleichsmatrix wurde erstellt.

### **Testfall Nr. 16**

- Bezeichnung: Aufbau des Smart-Home-Hub und verknüpfter Gateways
- Beschreibung: Überprüfe, ob ein Homeserver auf einem Raspberry Pi (z. B. über Home Assistant) konfiguriert wurde und ein Zigbee-Gateway sowie Hue-Geräte eingebunden wurden.
- Ziel: Aufbau des Smart-Home-Hub und der verknüpften Gateways
- Testschritte:
  1. Konfiguriere den Homeserver auf einem Raspberry Pi.
  2. Binde ein Zigbee-Gateway und Hue-Geräte ein.
- Erfüllungskriterium: Der Smart-Home-Hub und verknüpfte Gateways sind erfolgreich konfiguriert sowie eingebunden.

### **Testfall Nr. 17**

- Bezeichnung: Automatisierung/Abläufe
- Beschreibung: Überprüfe, ob Skripte zur Präsenzerkennung und zur automatischen Geräteabschaltung entwickelt, Routinen beim Verlassen und beim Betreten von Räumen oder bei der Wohnung erstellt sowie sprachgesteuerte Routinen für Events implementiert wurden.
- Ziel: Entwicklung von Automatisierungen und Abläufen.
- Testschritte:
  1. Teste die Präsenzerkennung und die automatische Geräteabschaltung.
  2. Teste Routinen bei Verlassen und Betreten von Räumen oder der Wohnung.
  3. Teste sprachgesteuerte Routinen für Events.
- Erfüllungskriterium: Automatisierungen und Abläufe sind erfolgreich entwickelt; sie funktionieren.

### **Testfall Nr. 18**

- Bezeichnung: grafische Oberfläche
- Beschreibung: Überprüfe, ob die grafische Benutzeroberfläche festgelegt und konfiguriert wurde.
- Ziel: Entwicklung der grafischen Oberfläche
- Testschritte:
  1. Überprüfe die Festlegung der Benutzeroberfläche.

2. Teste die Konfiguration der grafischen Oberfläche.

- Erfüllungskriterium: Die grafische Benutzeroberfläche ist erfolgreich festgelegt und konfiguriert.

**Testfall Nr. 19**

- Bezeichnung: globale Erreichbarkeit
- Beschreibung: Überprüfe, ob der Zugriff auf die grafische Oberfläche von überall mit einem Smartphone-Browser oder einer App ermöglicht wurde. Die Möglichkeit zur Steuerung von Licht und Steckdosen von überall auf der Welt wurde implementiert.
- Ziel: Implementierung der globalen Erreichbarkeit
- Testschritte:
  1. Versuche, von verschiedenen Orten aus auf die grafische Oberfläche zuzugreifen.
  2. Teste die Steuerung von Licht und Steckdosen von verschiedenen Orten aus.
- Erfüllungskriterium: Die globale Erreichbarkeit wurde erfolgreich implementiert und ermöglicht den Zugriff von verschiedenen Orten aus.

**Testfall Nr. 20**

- Bezeichnung: Zieldefinitionserfüllung
- Beschreibung: Überprüfe, ob das Smart-Home-System die in der Zieldefinition festgelegten Erfolgskriterien erfüllt.
- Ziel: Sicherstellen dessen, dass die festgelegten Erfolgskriterien erfüllt wurden
- Testschritte:
  1. Vergleiche die tatsächliche Leistung des Systems mit den festgelegten Erfolgskriterien.
  2. Stelle sicher, dass alle Erfolgskriterien erfüllt sind.
- Erfüllungskriterium: Das Smart-Home-System erfüllt alle festgelegten Erfolgskriterien.

**Testfall Nr. 21**

- Bezeichnung: Dokumentation der Ergebnisse
- Beschreibung: Überprüfe, ob die Ergebnisse der Entwicklung und der Tests gemäss den Anforderungen dokumentiert wurden.
- Ziel: Dokumentation der Ergebnisse
- Testschritte:
  1. Überprüfe die Vollständigkeit und die Richtigkeit der Dokumentation.
  2. Stelle sicher, dass alle entwickelten Funktionen und getesteten Aspekte dokumentiert sind.
- Erfüllungskriterium: Die Ergebnisse der Entwicklung und der Tests wurden vollständig sowie korrekt dokumentiert.

**Testfall Nr. 22**

- Bezeichnung: Benutzerfreundlichkeit
- Beschreibung: Überprüfe, ob die Oberfläche benutzerfreundlich ist und den Anforderungen entspricht.
- Ziel: Entwicklung einer benutzerfreundlichen Oberfläche
- Testschritte:
  1. Bewertung der Benutzerfreundlichkeit der grafischen Oberfläche

2. Prüfen dessen, ob die Benutzeroberfläche intuitiv und leicht verständlich ist

- Erfüllungskriterium: Die Benutzeroberfläche ist benutzerfreundlich und leicht verständlich.

**Testfall Nr. 23**

- Bezeichnung: automatische Lichtsteuerung
- Beschreibung: Überprüfe, ob die automatische Lichtsteuerung gemäss den Anforderungen funktioniert, basierend auf Anwesenheit und Bewegung im Raum.
- Ziel: Realisierung der automatischen Lichtsteuerung
- Testschritte:
  1. Überprüfe die automatische Ein- und Ausschaltung von Licht basierend auf Anwesenheit sowie Bewegung im Raum.
  2. Vergleiche die Aktionen mit den erwarteten Abläufen.
- Erfüllungskriterium: Die automatische Lichtsteuerung ist erfolgreich implementiert und entspricht den erwarteten Abläufen.

## 8.2.2 Auswertung der Testfälle

Testfall Nr.	Getestet am	Erkenntnisse	Erfüllungsgrad (%)
1	17.10.23	-	100
2	17.10.23	-	100
3	17.10.23	Standardbefehle funktionieren auf beiden Sprachsteuerungen einwandfrei	100
4	17.10.23	Steuerung über Smartswitches, Sprachsteuerung und Bewegungsmelder funktionieren einwandfrei	100
5	17.10.23	Temperatur stimmt ungefähr mit analogem Thermometer überein	100
6	18.10.23	Bedienung via Switches, Sprachsteuerung, Automatisierungen und App einwandfrei IR-Fernbedienung noch nicht lauffähig	80
7	18.10.23	Bewegungsmelder steuern wie gewünscht das Licht. Beim Verlassen des Hauses werden alle Lichter wie gewünscht ausgeschalten	100
8	18.10.23	Problemlose Steuerung mit den Schaltern	100
9	18.10.23	Ein und Ausschalten sowie Dimmen und Steuern von Geräten funktionieren einwandfrei	100
10	18.10.23	Statistiken zu allen Daten sind in den Dashboards einsehbar Stromverbrauch kann jedoch nicht angezeigt werden da die Sensoren nicht über diese Funktion verfügen Generation des Solarstrom wurde deswegen teilweise implementiert.	70
11	18.10.23	Erweiterbar mit Smart Schlössern und Bewegungsmeldung bei Abwesenheit	100
12	19.10.23	Intruder Alert und Movie Time funktionieren wie angedacht	100
13	19.10.23	Dashboards via Web und App erreichbar und funktions-tüchtig	100
14	19.10.23	Analyse Umgesetzt Da es aber ein stets veränderlicher Markt ist wird diese nicht auf Dauer Akkurat sein	100

15	19.10.23	Von den verwendeten Geräten alles IO Jedoch keine ausführliche Untersuchung aller Geräte da dies den Rahmen sprengen würde	70
16	19.10.23	ZigBee Gateway, Hue Geräte, Wifi Geräte, Shield Android TV, Sprachsteuerungen	100
17	19.10.23	Automatisierungen für Alle Use Cases abgedeckt und weiter hinzugefügt	100
18	19.10.23	Benutzerdefinierte Oberfläche ist erstellt worden Zukünftig werden andere Dashboard Plugins ausgetestet (Minimalistisch Design, etc)	100
19	19.10.23	Erreichbar weltweit via App und Web Ist notwendig für die Sprachassistenten	100
20	22.10.23	Leider konnten nicht ganz alle Kriterien erfüllt werden Ein Grund ist der IR Sensor und der andere ist das die Geräte vorgängig bestellt wurden und einige Features welche notwendig gewesen wären nicht beinhalteten  Erreichungsgrad durch die anderen 22 Testfällen berechnet	94
21	22.10.23	Dokumentation vorhanden und korrekt	100
22	19.10.23	-	100
23	19.10.23	Wird durch eine Automatisierung gehandhabt. Es wird aber nicht ein LUX wert ausgelesen, sondern mit dem Sonnen Plugin und Ruhezeiten gearbeitet	75

Tabelle 7 Protokoll der Testfälle

### 8.3 Reflexion des Wegs zum Ziel

Wie erwähnt waren meine privaten Verpflichtungen einer guten, schnellen und effektiven Umsetzung des gesamten Projektes nicht förderlich.

Den Austausch mit der Betreuungsperson, welcher auch mein Experte ist, fand ich wertvoll. Er hat mir geholfen und mich motiviert. Dies ermöglichte es mir, die Projektumsetzung wie geplant voranzutreiben.

Zudem habe ich während der Umsetzung wiederholt mit meinen Freunden und Eltern über mein Projekt gesprochen bzw. habe ich es ihnen gezeigt. Im Rahmen dieses Testens konnte ich einige Bugs erkennen und beheben.

Der Stress aus dem Arbeitstag hat mich deutlich beeinflusst. Ich war immer wieder müde, gestresst und spät zuhause, was mich vom Schreiben abgehalten hat. Als Massnahme dagegen habe ich einige Tage im Geschäft kompensiert, um mehr Zeit zu erhalten und mich zu entlasten.

Im Allgemeinen war meine Herangehensweise zielführend. Könnte ich die Bearbeitung neu beginnen, würde ich mich im Voraus umfassender informieren, damit ich besser passende Geräte

einkaufen kann. Mit dem aktuellen Endprodukt kann ich weiterarbeiten. Ich kann den Prototyp weiterentwickeln, weshalb ich ihn positiv wahrnehme.

#### **8.4 Lessons learned**

Dank des Projekts habe ich gelernt, besser zu planen. Ich habe die Relevanz dessen erkannt, von Anfang an einen Plan zu befolgen, da dieser als Stütze dient. Dennoch ist aufgrund der stets auftretenden unvorhergesehenen Änderungen auch Flexibilität nötig.

Anhand des Endprodukts habe ich viele verschiedene Technologien und Informationen kennengelernt, aber auch vorhandenes Wissen gut anwenden können. Es hat mir gefallen, verschiedene Technologien zu verstehen und ausprobieren zu können. Ich möchte weitere Alternativen erkunden.

Ferner habe ich gelernt, dass ein regelmässiges Prüfen dessen nötig ist, ob die Hardware wie geplant funktioniert. Es gilt, im Voraus Alternativen und Back-up-Lösungen zu definieren.

Zudem habe ich nicht von Anfang an bedacht, dass ich nicht der Einzige bin, der gewisse Funktionen zu implementieren versucht. Ich hätte die Community somit früher zum Thema befragen können, um nicht alle Probleme selbst lösen zu müssen.

Zudem wurde mir bewusst, wie viele Geräte mittlerweile netzfähig sind und was mit simplen Automatisierungen sowie Auswertungen erreichbar ist.

Als zentrale Erkenntnis nehme ich mit, nur wenig auf einmal zu planen sowie Reserven zu definieren und zu nutzen. Der Austausch mit anderen hilft, neue Perspektiven einzunehmen und eventuell Lösungen für eigene Probleme zu finden.

#### **8.5 Ausblick**

Während der Arbeit habe ich viele neue Geräte und Informationen gefunden, die sich dafür eignen, Automationen auszubauen oder zu verbessern. Diese werden in den nächsten paar Wochen angeschafft, ausprobiert und integriert. Ich möchte den Gesamtverbrauch des Hauses messen können. Die Heizung ist auch smart und soll integriert werden, was aber noch nicht erfolgte, da die Log-in-Daten fehlen. Die Netzwerküberwachung soll ausgebaut werden, unter anderem weil mein Vater ein Geschäft zuhause hat und wir nun das Netz umbauen, damit dieses vom Heimnetzwerk isoliert wird.

Die Infrarot-Fernsteuerung, welche noch nicht funktioniert, soll korrigiert werden oder es soll eine Alternative implementiert werden. Viele Ideen sind vorhanden und es dürften weitere hinzukommen.

## Abkürzungsverzeichnis

API *Application Programming Interface*  
ARN *Amazon Resource Name*  
AWS *Amazon Web Service*  
DHCP *Dynamic Host Configuration  
Protocol*  
DIY *Do-it-Yourself*  
DNS *Domain Name System*  
GUI *Graphical User Interface*  
HA *Home Assistant*  
IEEE *Institute of Electrical and Electronics  
Engineers*  
IoT *Internet of Things*  
IP *Internetprotokoll*  
ISO *International Organization for  
Standardization*

KI *Künstliche Intelligenz*  
MAC *Medium-Access-Control*  
OS *Operating System*  
PAP *Projektablaufplan*  
PHY *physikalische Schicht*  
PoC *Proof of Concept*  
PSP *Projektstrukturplanung*  
SD *Secure Digital Memory*  
SSH *Secure Shell*  
SSL *Secure Sockets Layer*  
URI *Uniform Resource Identifier*  
WiFi *Wireless Fidelity*  
WLAN *Wireless Local Area Network*

## Abbildungsverzeichnis

Abbildung 1 Lebenslauf.....	4
Abbildung 2 Projektstrukturplan.....	9
Abbildung 3 Projektablaufplan Soll Zustand .....	10
Abbildung 4 IST Planung - September .....	10
Abbildung 5 IST Planung - Oktober .....	11
Abbildung 6 Basis-Architekturplan .....	19
Abbildung 7 Detail Architektur Ansicht .....	20
Abbildung 8 Beispiel der Dashboard-Ansicht .....	25
Abbildung 9 Vorschaubild für Integrationen .....	26
Abbildung 10 Vorschau der Geräte- und Entitätenkonfiguration .....	27
Abbildung 11 GUI für die Automationserstellung.....	27
Abbildung 12 Vorschau der Skriptkonfiguration.....	28
Abbildung 13 Vorschau des Szenen-Editor .....	29
Abbildung 14 Vorschau der Add-on-Übersicht .....	29
Abbildung 15 Home Assistant Feature-List.....	30
Abbildung 16 Pi-Imager-OS-Auswahl.....	31
Abbildung 17 Pi Imager – Schreiben des OS auf die SD-Karte.....	31
Abbildung 18 Vorschau des Willkommens-Bildschirms.....	32
Abbildung 19 User-Set-up .....	33
Abbildung 20 Informationsteilung .....	34
Abbildung 21 Fritz!Box-User-Set-up.....	35
Abbildung 22 Konfiguration des HA-Zugriffs auf die Fritz!Box .....	35
Abbildung 23 Fritz!Box-Port-Konfiguration .....	36
Abbildung 24 Weg zum Add-on-Store .....	37
Abbildung 25 Add-on-Store-Übersicht.....	38
Abbildung 26 Übersicht des DuckDNS-Add-on.....	39
Abbildung 27 Konfiguration von DuckDNS.....	40
Abbildung 28 Starten des DuckDNS-Add-on .....	40
Abbildung 29 Konfigurationsdatei anpassen .....	41
Abbildung 30 Neustart des HA-Systems .....	41
Abbildung 31 Tuya Steckdose im Büro – Schreibtisch .....	42
Abbildung 32 Zigbee Steckdose für Entertainment – Schlafzimmer.....	42
Abbildung 33 Philipps Steckdose Nachttisch – Schlafzimmer .....	43
Abbildung 34 Tür- und Fenstersensor.....	43
Abbildung 35 Temperatursensor – Schlafzimmer.....	43
Abbildung 36 Magic Home Lampe und Zigbee Bewegungsmelder – Schlafzimmer.....	44
Abbildung 37 Bewegungsmelder im Büro.....	44
Abbildung 38 Schalter vor der Demontage.....	44
Abbildung 39 Eingebauter Sonoff Switch mit Änderung zu einfachem Stecker.....	45
Abbildung 40 Endresultat des Schalterumbaus.....	45
Abbildung 41 Smartswitch Einbau direkt bei der Lampe .....	46

Abbildung 42 Zigbee-Integration.....	46
Abbildung 43 Zigbee-Integration Konfigurationsübersicht.....	47
Abbildung 44 Zigbee-Geräte hinzufügen.....	47
Abbildung 45 Zigbee-Paarungsmodus.....	47
Abbildung 46 ZigBee-Gerät ermittelt.....	48
Abbildung 47 Zigbee-Gerät umbenennen und Bereichszuordnung.....	48
Abbildung 48 Tuya-Cloud-Projekterstellung.....	49
Abbildung 49 Tuya-API-Konfiguration.....	49
Abbildung 50 Tuya-Account-Link.....	50
Abbildung 51 Tuya-Link-QR-Code.....	50
Abbildung 52 Tuya-Device-Übersicht.....	50
Abbildung 53 Tuya Konfigurationsinformationen.....	51
Abbildung 54 HA Tuya Integrationskonfiguration.....	51
Abbildung 55 Upload des Account-Files.....	53
Abbildung 56 Übersicht der Automatisierungen.....	56
Abbildung 57 Erstellung neuer Automation.....	57
Abbildung 58 Automationsansicht zum YAML Editor wechseln.....	57
Abbildung 59 Standard-Dashboard.....	59
Abbildung 60 Remote-Dashboard.....	60
Abbildung 61 Schlafzimmer-Dashboard.....	60
Abbildung 62 Büro-Dashboard.....	61
Abbildung 63 Allerlei-Dashboard mit Testkarte.....	61
Abbildung 64 Energie-Übersicht.....	62
Abbildung 65 Statusbericht Eins.....	86
Abbildung 66 Statusbericht Zwei.....	86
Abbildung 67 Statusbericht Drei.....	87
Abbildung 68 Statusbericht Vier.....	87
Abbildung 69 Statusbericht Fünf.....	88
Abbildung 70 Statusbericht Sechs.....	88
Abbildung 71 Anhang 3 - configuration.yaml.....	89

## Tabellenverzeichnis

Tabelle 1 Risikoanalyse .....	11
Tabelle 2 Smart-Home-Hubs im Vergleich .....	13
Tabelle 3 Präferenzmatrix .....	17
Tabelle 4 Übersicht der Smart Devices von Aliexpress .....	22
Tabelle 5 Übersicht der Smart Devices von Digitec .....	23
Tabelle 6 Vorhandene Produkte .....	24
Tabelle 7 Protokoll der Testfälle .....	71

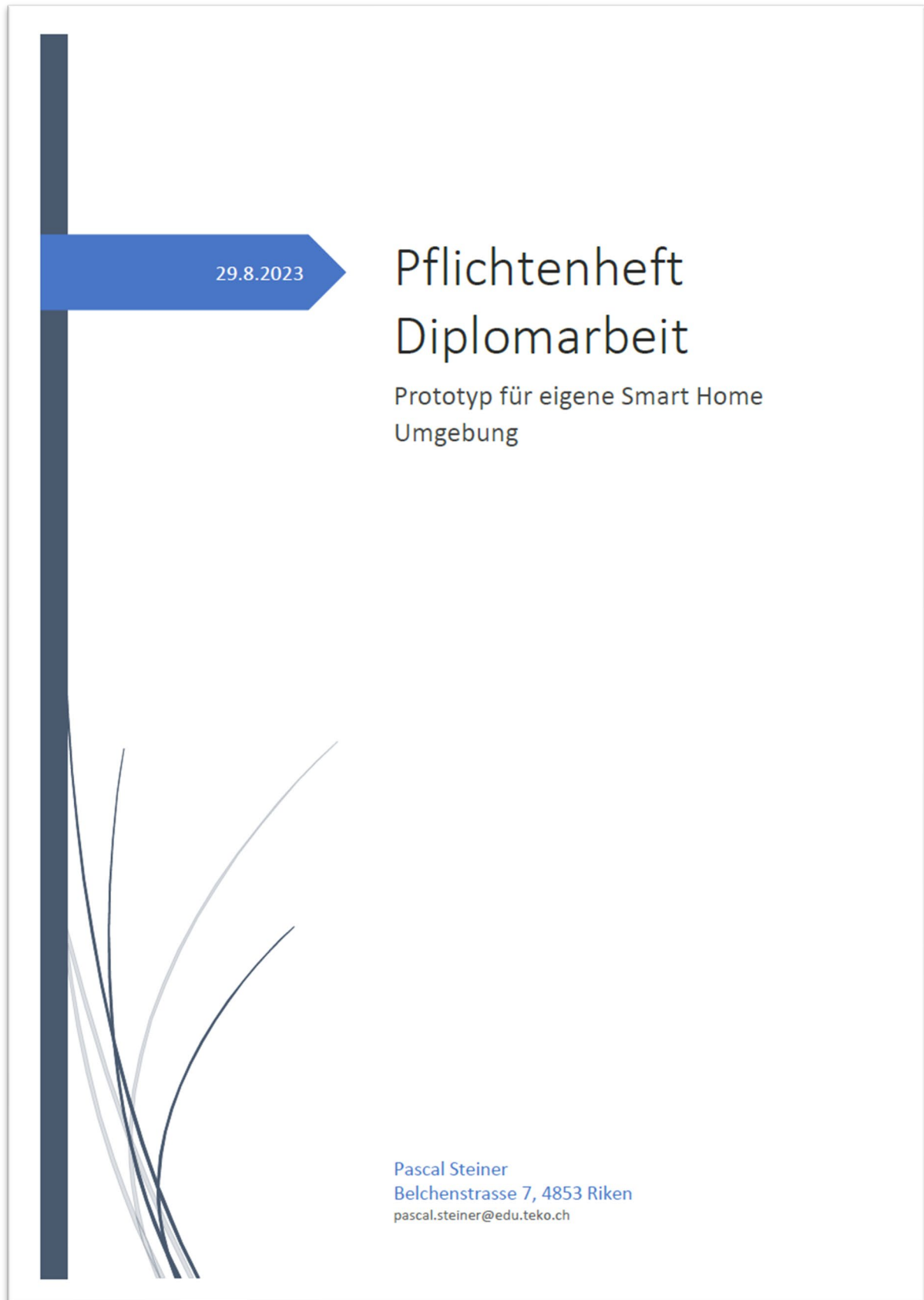
## Literatur- und Quellenverzeichnis

- Aliexpress Online Shop*. (2023). Von Aliexpress.com: <https://de.aliexpress.com/> abgerufen
- Amazon Echo & Alexa*. (2023). Von Amazon:  
<https://www.amazon.de/b?ie=UTF8&node=14100226031> abgerufen
- Apple HomeKit*. (2023). Von Apple: [https://support.apple.com/de-de/HT204893#:~:text=Schalte%20dein%20Ger%C3%A4t\\*%20ein%2C%20und,des%20Zubeh%C3%B6rs%2C%20um%20es%20hinzuzuf%C3%BCgen](https://support.apple.com/de-de/HT204893#:~:text=Schalte%20dein%20Ger%C3%A4t*%20ein%2C%20und,des%20Zubeh%C3%B6rs%2C%20um%20es%20hinzuzuf%C3%BCgen). abgerufen
- Apple HomeKit Accessories*. (2023). Von Apple: <https://www.apple.com/ch-de/shop/accessories/all/homekit> abgerufen
- Bradford, A. (2023). *Safewise What's the Difference between Zigbee and Z-Wave?* Von Safewise:  
<https://www.safewise.com/blog/zigbee-vs-zwave-review/> abgerufen
- ChatGPT*. (2023). Von Chat OpenAI 3.5: <https://chat.openai.com/> abgerufen
- CNET Smart Home* . (2023). Von CNET: <https://www.cnet.com/home/smart-home/> abgerufen
- Conectivity standarts alliance Zigbee*. (2023). Von CSA-iot: <https://csa-iot.org/all-solutions/zigbee/> abgerufen
- David Priest, R. B. (September 2023). *CNET Best Smart Home Devices*. Von CNET:  
<https://www.cnet.com/home/smart-home/best-smart-home-devices/> abgerufen
- Digitec Online Shop*. (2023). Von Digitec.ch: <https://www.digitec.ch/> abgerufen
- Google Console*. (2023). Von Google: <https://console.actions.google.com> abgerufen
- Google Home*. (2023). Von Google.com: <https://home.google.com/welcome/> abgerufen
- Google Store Smart Home*. (2023). Von Google.com:  
[https://store.google.com/de/category/connected\\_home?hl=de&pli=1](https://store.google.com/de/category/connected_home?hl=de&pli=1) abgerufen
- Home Assistant* . (Oktober 2023). Von Home Assistant: <https://www.home-assistant.io/> abgerufen
- Home Assistant Community*. (5. Oktober 2023). Von Home Assistant: <https://community.home-assistant.io/> abgerufen
- Meredith Hirt, S. A. (2023). *Forbes Smart Home Trends* . Von Forbes:  
<https://www.forbes.com/home-improvement/internet/smart-home-tech-trends/> abgerufen
- OpenHAB*. (2023). Von Openhab.org: <https://www.openhab.org/> abgerufen
- Pastebin MaxComm\_Protocol\_Explanation.txt*. (August 2023). Von Pastebin :  
<https://pastebin.com/SwpEzdsZ> abgerufen
- Push, A. (Februar 2021). *CNET The Pros and cons of a smart home hub*. Von CNET.com:  
<https://www.cnet.com/home/smart-home/the-pros-and-cons-of-a-smart-home-hub/> abgerufen
- Raspberry Pi*. (2023). Von Raspberrypi.com: <https://www.raspberrypi.com/software/> abgerufen
- Samsung Smartthings*. (2023). Von Samsung: <https://www.samsung.com/ch/apps/smartthings/> abgerufen
- Schweizerisch Fachschule Teko* . (2023). Von Teko : <https://www.teko.ch/> abgerufen
- Statista Smart Home - Weltweit*. (September 2023). Von Statista:  
<https://de.statista.com/outlook/dmo/smart-home/weltweit> abgerufen
- Taylor, P. (Mai 2023). *Statista Smart home market revenue worldwide*. Von Statista:  
<https://www.statista.com/statistics/823663/smart-home-market-revenue-worldwide-by-region/> abgerufen

- TheHellSite. (2023). *Community HomeAssistant*. Von Home Assistant: <https://community.home-assistant.io/t/tutorial-solarmax-s-series-inverter-integration/608464> abgerufen
- Thormundsson, B. (April 2023). *Statista Smart home - statistics & facts*. Von Statista: <https://www.statista.com/topics/2430/smart-homes/#editorsPicks> abgerufen
- Wikipedia Matter Sandart*. (30. September 2023). Von Wikipedia : [https://de.wikipedia.org/wiki/Matter\\_\(Standard\)](https://de.wikipedia.org/wiki/Matter_(Standard)) abgerufen
- YouTube*. (2023). Von Youtube.com: [https://www.youtube.com/results?search\\_query=home+assistant](https://www.youtube.com/results?search_query=home+assistant) abgerufen
- Z-Wave Alliance* . (2023). Von Z-Wave Alliance : <https://z-wavealliance.org/> abgerufen

## Anhang

### 8.6 Anhang 1: Pflichtenheft



Pflichtenheft Diplomarbeit

**TEKO**

### Inhaltsverzeichnis

Pflichtenheft Diplomarbeit Pascal Steiner .....	2
1. Einleitung.....	2
2. Fachexperte .....	2
3. Inhalt.....	2
3.1 Richtziel .....	2
3.2 Ziele und Erfolgskriterien .....	3
3.3 Technische Anforderungen und schematische Skizzen .....	4
3.4 Schematische Skizzen und Diagramme .....	5
3.5 Weitere Angaben .....	5
Abschließende Bemerkungen.....	6
Freigabe des Pflichtenhefts: .....	6

Pascal Steiner

1

## Pflichtenheft Diplomarbeit Pascal Steiner

### 1. Einleitung

Die vorliegende Diplomarbeit beschäftigt sich mit der Entwicklung und Realisierung einer individuellen Smart-Home-Lösung unter Verwendung von verschiedenen Technologien wie Home Assistant, Zigbee, Hue, Google Assistant, Alexa und weiteren Integrationen. Ziel dieser Arbeit ist es, einen funktionsfähigen Prototypen zu erstellen, der als Grundlage für die Einrichtung einer zukünftigen Wohnung mit smarten Systemen dient. Dies beinhaltet auch eine detaillierte Analyse aller zur verfügbaren Zigbee Sensoren und deren Auswertungsmatrix. Sowie eine Marktanalyse der verfügbaren Smarthome Hub Lösungen.

Die aktuelle Marktsituation bietet bereits eine Vielzahl von Smart-Home-Produkten und Lösungen. Allerdings sind diese oft auf spezifische Anwendungen beschränkt oder erfordern umfangreiche Anpassungen, um den individuellen Bedürfnissen gerecht zu werden. Durch die Entwicklung einer maßgeschneiderten Smart-Home-Lösung strebt diese Arbeit danach, eine flexible und erweiterbare Plattform zu schaffen, die eine nahtlose Integration und Steuerung verschiedener Geräte und Dienste ermöglicht.

### 2. Fachexperte

Benjamin Bäni hat mir versichert, dass für die Umsetzung dieser Arbeit ist kein spezifischer Fachexperte erforderlich ist. Die verwendeten Technologien und Konzepte werden im Verlauf der Arbeit ausführlich erläutert und dokumentiert.

### 3. Inhalt

#### 3.1 Richtziel

Das Richtziel dieser Diplomarbeit ist es, einen funktionsfähigen Prototypen einer selbstgebauten Smart-Home-Lösung zu entwickeln und zu dokumentieren. Der Prototyp soll die nahtlose Kommunikation und Steuerung verschiedener Geräte und Dienste über eine zentrale Plattform ermöglichen. Die erarbeitete Lösung soll als Leitfaden für die Einrichtung eines realen Smart Homes dienen und dabei eine effiziente Nutzung von Energie und Ressourcen fördern. Sie soll zudem eine detaillierte Analyse und Auswertungsmatrix beinhalten.

## Pflichtenheft Diplomarbeit



## 3.2 Ziele und Erfolgskriterien

Die detaillierten Ziele und Erfolgskriterien dieser Diplomarbeit sind wie folgt definiert:

**Aufbau eines funktionsfähigen Prototyps:** Entwicklung und Zusammenstellung eines Prototyps eines Smart-Home-Systems, das verschiedene Komponenten und Technologien integriert, darunter Zigbee-fähige Geräte, Beleuchtungssteuerung, Temperaturregelung und Sicherheitssysteme.

**Implementierung grundlegender Funktionalitäten:** Realisierung von Schlüsselfunktionen, darunter die Fernsteuerung der Beleuchtung, die Überwachung und Anpassung der Raumtemperatur sowie die Einbindung von Sicherheitskomponenten wie Kameras und Bewegungsmeldern.

**Integration von Zigbee-Geräten:** Einbindung von mindestens drei verschiedenen Zigbee-fähigen Geräten verschiedener Hersteller, um die Interoperabilität und das Zusammenspiel unterschiedlicher Technologien zu demonstrieren.

**Sprachsteuerung über Google Assistant und Alexa:** Konfiguration und Integration von Google Assistant und Alexa, um eine intuitive Sprachsteuerung der Smart-Home-Geräte zu ermöglichen.

**Entwicklung einer Benutzeroberfläche:** Erstellung einer benutzerfreundlichen Benutzeroberfläche, die es den Nutzern ermöglicht, das Smart Home zu überwachen, zu steuern und Einstellungen anzupassen.

**Durchführung umfassender Tests:** Gründliche Erprobung des Prototyps unter verschiedenen Szenarien und Anwendungsfällen, um die Funktionalität, Zuverlässigkeit und Interoperabilität sicherzustellen.

**Dokumentation der Ergebnisse:** Umfassende Dokumentation der Systemarchitektur, der verwendeten Technologien, der Implementierungsdetails, der Schnittstellen und der durchgeführten Tests.

Nachfolgend nochmals alle gewünschten Endergebnisse und deren Erfolgsmessung vereinfacht zusammengestellt:

- **Generelle Marktanalyse:**
  - Analyse über DIY (Do it Yourself) Smart-Home-Geräte.
  - Beweggründe für die gewählte Lösung darstellen.
- **Sensoren und Aktoren Analyse und Vergleichsmatrix:**
  - Untersuchung verschiedener Sensor- und Aktortypen.
  - Erstellung einer Vergleichsmatrix mit Vor- und Nachteilen.
  - Dokumentation der Integration aller verfügbaren Sensoren und Aktoren.
- **Aufbau des Smart-Home-Hub und verknüpfter Gateways:**
  - Konfiguration eines Homeservers auf einem Raspberry Pi (z.B. HomeAssistant).
  - Integration eines Zigbee-Gateways.
  - Einbindung einer Hue Bridge.
  - Konfiguration von Google Assistant, Amazon Alexa und anderer Standalone WLAN Devices.
- **Automatisierung/Abläufe Skripte erstellen:**
  - Entwicklung von Skripten zur Präsenzerkennung und automatischen Geräteabschaltung.
  - Erstellung von Routinen bei Verlassen und Betreten von Räumen oder der Wohnung.
  - Implementierung sprachgesteuerter Routinen für Events.
- **Grafische Oberfläche:**

## Pflichtenheft Diplomarbeit



- Analyse bestehender Tools.
- Festlegung einer Benutzeroberfläche.
- Konfiguration der grafischen Oberfläche.
- Geschützte Globale Erreichbarkeit:
  - Zugriff auf die grafische Oberfläche von überall mittels Smartphone-Browser oder einer App.
  - Möglichkeit zur Steuerung von Licht und Steckdosen von überall auf der Welt.

Aufgrund das diese Arbeit für einen Privaten Rahmen ist werden nur DIY-Lösungen berücksichtigt und Keine Professionellen Installationen, welche ein grösseres Budget beinhaltet. Die Arbeit sollte nachträglich Jedermann nutzen können, um somit günstig und einigermaßen schnell seine Wohnung selbst zu einer Smarten Umgebung umzurüsten zu können.

### 3.3 Technische Anforderungen und schematische Skizzen

Die technischen Anforderungen für den Prototypen umfassen:

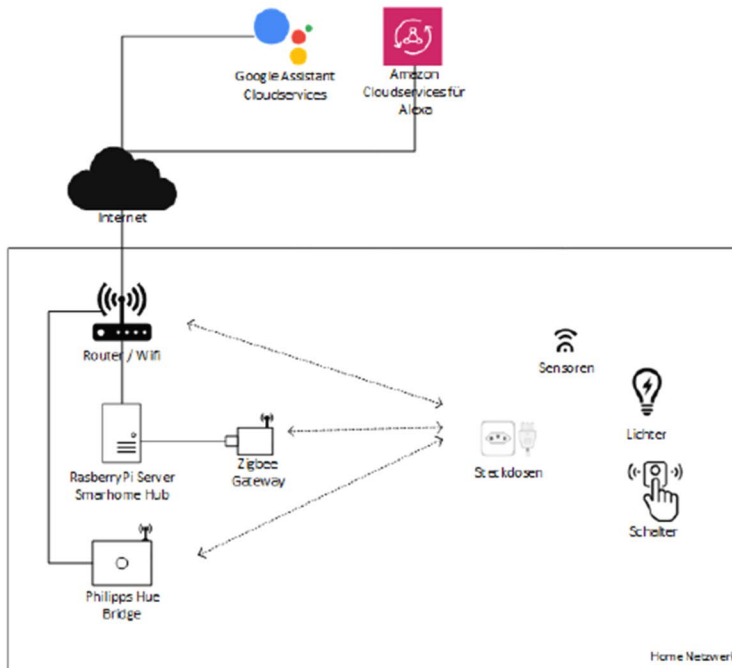
- Kompatibilität und Integration:
  - Gewährleistung der nahtlosen Integration und Kommunikation zwischen verschiedenen Komponenten, einschließlich Zigbee, Hue, Google Assistant und Alexa.
- Sicherheit und Datenschutz:
  - Implementierung von Sicherheitsmaßnahmen zur Gewährleistung der Vertraulichkeit und Integrität von Daten und zur Verhinderung unbefugter Zugriffe.
- Modularer Aufbau:
  - Strukturierung des Systems in Module, um eine einfache Erweiterung und Anpassung für zukünftige Anforderungen zu ermöglichen.
- Energieeffizienz:
  - Berücksichtigung von Energieeffizienzmaßnahmen, um den Energieverbrauch der Smart-Home-Geräte zu optimieren.
- Benutzerfreundlichkeit:
  - Gestaltung einer intuitiven Benutzeroberfläche, die es den Nutzern ermöglicht, das Smart Home einfach zu bedienen und zu überwachen.

Pflichtenheft Diplomarbeit



3.4 Schematische Skizzen und Diagramme

Für eine klare Darstellung der Systemarchitektur, der Teilsysteme und der Schnittstellen wurde nachfolgende eine grobe Darstellung erzeugt. Diese visualisieren den Aufbau des Smart-Home-Prototyps.



Im Rahmen der Arbeit werden detaillierte Schemas und Diagramme erstellt, um die Systemarchitektur, Teilsysteme und Schnittstellen klarer darzustellen.

3.5 Weitere Angaben

Die enge Zusammenarbeit mit dem Betreuer und/oder dem Fachexperten (falls notwendig) gewährleistet, dass die gesteckten Ziele erreicht werden und die erstellte Smart-Home-Lösung den höchsten Qualitätsstandards entspricht.

Da es sich bei dieser Diplomarbeit um ein privates Projekt handelt, werden keine firmenspezifischen Vorlagen verwendet. Die Arbeit orientiert sich an den Formvorgaben der Teko.

Pflichtenheft Diplomarbeit

**TEKO**

### Abschließende Bemerkungen

Die vorliegende Diplomarbeit strebt die Entwicklung einer "Ready to Deploy" Lösung an. Lediglich die Integration weiterer Sensoren, Aktoren und Raumkonfigurationen ist erforderlich. Die Arbeit ist so strukturiert, dass jeder anhand dieser Anleitung das gleiche Setup erstellen kann. Dabei werden ausschließlich DIY-Lösungen berücksichtigt, da diese kostengünstig und schnell für den Selbstbau einer Smart-Home-Umgebung genutzt werden können.

### Freigabe des Pflichtenhefts:

Freigegeben durch:

Ort/Datum

Signatur

Moskowan, 10.9.2023 R. Rami

Pascal Steiner

6

## 8.7 Anhang 2: Projektstatusberichte

### Projekt: Eigene Smart -Home Lösung

Stautsbericht: 1 - KW 38 - 18.9.23

<b>Projektleiter</b> Pascal Steiner	<b>Projektziele</b> Eigene Smarthome Umgebung erstellen und bereitstellen. Markt vergleich bekannter und gängiger SmartHome Hubs und Gadgets sowie Technologien.	<b>Verteiler</b> • Benjamin Bani			
<b>Gesamtbeurteilung</b>	<b>Projektverlauf</b> 	<b>Projektklima</b> 	<b>Termine</b> 	<b>Risiken</b> 	<b>Ressourcen</b> 
<b>Tendenz</b>					
<b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Dokumentationsvorlage erstellt</li> <li>• Projektplanung erstellt</li> <li>• Erste Annäherungen an HomeAssistant durchgeführt</li> </ul>		<b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Die Dokumenten Struktur konnte gut erarbeitet werden. Erste Testläufe mit Home Assistant sind erfolgreich erfolgt.</li> <li>• Konnte schon viele Dokumentationen und Informationen bezüglich HomeAssistant (HA oder HAS) finden und lesen</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Ein Sensor ist Defekt, Der Infrarot Sensor muss manuell konfiguriert und erlernt werden.</li> </ul>			
<b>Geplante nächste Schritte / getroffene Massnahmen</b> <ul style="list-style-type: none"> <li>• Markt analyse/vergleich erstellen und Technologien erläutern.</li> <li>• Mehr Zeit investieren und erste Vorbereitung für Experten Gespräch erstellen (Erste Woche konnte nicht viel umgesetzt werden, wegen persönlicher Termine und Abwesenheiten)</li> </ul>					

ProjektStatusbericht; Stefan Thöni, Josef Rüber

Abbildung 65 Statusbericht Eins

### Projekt: Eigene Smart -Home Lösung

Stautsbericht: 2 - KW 39 - 25.9.23

<b>Projektleiter</b> Pascal Steiner	<b>Projektziele</b> Eigene Smarthome Umgebung erstellen und bereitstellen. Markt vergleich bekannter und gängiger SmartHome Hubs und Gadgets sowie Technologien.	<b>Verteiler</b> • Benjamin Bani			
<b>Gesamtbeurteilung</b>	<b>Projektverlauf</b> 	<b>Projektklima</b> 	<b>Termine</b> 	<b>Risiken</b> 	<b>Ressourcen</b> 
<b>Tendenz</b>					
<b>Aktueller Projektstand</b> <ul style="list-style-type: none"> <li>• Dokumentationsvorlage komplett überarbeitet</li> <li>• PSP und Ablauf erstellt und überarbeitet</li> <li>• Qualifikationsprofil erstellt und Ausgangslage beschrieben</li> <li>• Analyse gestartet</li> </ul>		<b>Was läuft gut?</b> <ul style="list-style-type: none"> <li>• Die Dokumentation kommt voran da nun auch mehr informationer durch das gespräch klar wurde</li> </ul> <b>Was läuft nicht gut?</b> <ul style="list-style-type: none"> <li>• Es war nicht gut das ich noch Pikettdienst hatte, dies hat mich in ablauf wieder etwas verzögert.</li> </ul>			
<b>Geplante nächste Schritte / getroffene Massnahmen</b> <ul style="list-style-type: none"> <li>• Analyse/vergleich fertigstellen.</li> <li>• Werde ein paar kompensations tage beiziehen um die verlorene Zeit aufzuholen.</li> </ul>					

ProjektStatusbericht; Stefan Thöni, Josef Rüber

Abbildung 66 Statusbericht Zwei

**Projekt: Eigene Smart -Home Lösung**

**Stautsbericht: 3 - KW 40 - 02.10.23**

<b>Projektleiter</b> Pascal Steiner	<b>Projektziele</b> Eigene Smarthome Umgebung erstellen und bereitstellen. Markt vergleich bekannter und gängiger SmartHome Hubs und Gadgets sowie Technologien.	<b>Verteiler</b> • BenjaminBäni			
<b>Gesamtbeurteilung</b>	<b>Projektverlauf</b> □ □ □	<b>Projektklima</b> □ □ □	<b>Termine</b> □ □ □	<b>Risiken</b> □ □ □	<b>Ressourcen</b> □ □ □
<b>Tendenz</b>	⇒	⇒	⇒	⇒	⇒
<b>Aktueller Projektstand</b>		<b>Was läuft gut?</b>			
<ul style="list-style-type: none"> <li>Analyse Abgeschlossen</li> <li>Initialisierung Abgeschlossen</li> <li>Vorhandene Sensoren integriert und eingerichtet</li> <li>HomeAssistant exposed zum Internet</li> <li>Automatisierungen Begonnen</li> <li>Integration mit Alexa und Google Home begonnen</li> </ul>		<ul style="list-style-type: none"> <li>Die Dokumentation kommt stetig voran.</li> <li>Bisher keine weiteren Defektergeräte</li> </ul> <p><b>Was läuft nicht gut?</b></p> <ul style="list-style-type: none"> <li>Habe 2 meiner gesetzten Meilensteine noch nicht erreicht.</li> <li>Musste Umgehungslösung für einig SmartSwitcheseinrichten.</li> </ul>			
<b>Geplante nächste Schritte / getroffene Massnahmen</b>					
<ul style="list-style-type: none"> <li>Bin in Abklärung ob ich diese Woche auch noch einen Kompensation Tag einlegen kann damit ich die verpassten Meilensteine aufholen kann. Sollte meiner Meinung nach mit einem Tag erreichbar sein.</li> <li>Sprachassistenten fertig integrieren, alle definierten Automationen fertigstellen und dann Testfälle erstellen</li> </ul>					

Projektstatusbericht; Stefan Thöni, Josef Räber

Abbildung 67 Statusbericht Drei

**Projekt: Eigene Smart -Home Lösung**

**Stautsbericht: 4 - KW 41 - 9.10.23**

<b>Projektleiter</b> Pascal Steiner	<b>Projektziele</b> Eigene Smarthome Umgebung erstellen und bereitstellen. Markt vergleich bekannter und gängiger SmartHome Hubs und Gadgets sowie Technologien.	<b>Verteiler</b> • BenjaminBäni			
<b>Gesamtbeurteilung</b>	<b>Projektverlauf</b> □ □ □	<b>Projektklima</b> □ □ □	<b>Termine</b> □ □ □	<b>Risiken</b> □ □ □	<b>Ressourcen</b> □ □ □
<b>Tendenz</b>	⇒	⇒	⇒	⇒	⇒
<b>Aktueller Projektstand</b>		<b>Was läuft gut?</b>			
<ul style="list-style-type: none"> <li>Alles Dokumentiert was aktuell möglich ist</li> <li>Funktionierende Automationen und Homeassistant Server</li> <li>Alle Integrationen gemacht.</li> </ul>		<ul style="list-style-type: none"> <li>Alles, was geht ist nachdokumentiert.</li> <li>Sensor konnte repariert werden.</li> </ul> <p><b>Was läuft nicht gut?</b></p> <ul style="list-style-type: none"> <li>Bringe den Infrarot Sensor nicht zum Laufen, dem entsprechen gehen seine Automationen nicht.</li> <li>Bin leider immer noch nicht mit meinem Zeitplan überein werde das Lektorieren verkürzen müssen.</li> </ul>			
<b>Geplante nächste Schritte / getroffene Massnahmen</b>					
<ul style="list-style-type: none"> <li>Alexa Integration muss ich noch erledigen.</li> <li>Testing und deren Protokolle muss noch Sauberdurchgeführt werden</li> <li>Und dann noch den Abschluss schreiben</li> </ul>					

Projektstatusbericht; Stefan Thöni, Josef Räber

Abbildung 68 Statusbericht Vier

**Projekt: Eigene Smart -Home Lösung**

**Stautsbericht: 5 - KW 42 - 16.10.23**

<b>Projektleiter</b> Pascal Steiner	<b>Projektziele</b> Eigene Smarthom&Umgebung erstellen und bereitstellen. Markt vergleich bekannter und gängiger SmartHome Hubs und Gadgets sowie Technologien.	<b>Verteiler</b> • BenjaminBäni			
<b>Gesamtbeurteilung</b>	<b>Projektverlauf</b> ■ □ □	<b>Projektklima</b> ■ □ □	<b>Termine</b> □ ■ □	<b>Risiken</b> ■ □ □	<b>Ressourcen</b> ■ □ □
<b>Tendenz</b>	⇒	⇒	⇒	⇒	⇒
<b>Aktueller Projektstand</b>			<b>Was läuft gut?</b>		
<ul style="list-style-type: none"> <li>Praktische Arbeit des Projekts abgeschlossen</li> <li>Layout und LetzteDokumentation Anpassungen sind noch offen.</li> <li>Lektorieren ausstehend</li> </ul>			<ul style="list-style-type: none"> <li>Diplomarbeit ist im Endspurt und somit fast fertig.</li> </ul>		
			<b>Was läuft nicht gut?</b>		
			<ul style="list-style-type: none"> <li>Hatte noch weitere Verzögerung aber konnte durch Reserve in der Planung abgedefert werden.</li> <li>Layout passt noch nicht und Zeitformen stimmen noch nicht überall.</li> </ul>		
<b>Geplante nächste Schritte / getroffene Massnahmen</b>					
<ul style="list-style-type: none"> <li>Vorbereitung für Lektorieren</li> <li>Endüberarbeitung und Feedback einfließen lassen</li> <li>Bilder, Diagramme, Pläne überprüfen aktualisieren und ergänzen</li> </ul>					

ProjektStatusbericht; Stefan Thöni, Josef Räber

Abbildung 69 Statusbericht Fünf

**Projekt: Eigene Smart -Home Lösung**

**Stautsbericht 6 - KW 43 - 23.10.23**

<b>Projektleiter</b> Pascal Steiner	<b>Projektziele</b> Eigene Smarthom&Umgebung erstellen und bereitstellen. Markt vergleich bekannter und gängiger SmartHome Hubs und Gadgets sowie Technologien.	<b>Verteiler</b> • BenjaminBäni			
<b>Gesamtbeurteilung</b>	<b>Projektverlauf</b> ■ □ □	<b>Projektklima</b> ■ □ □	<b>Termine</b> ■ □ □	<b>Risiken</b> ■ □ □	<b>Ressourcen</b> ■ □ □
<b>Tendenz</b>	⇒	⇒	⇒	⇒	⇒
<b>Aktueller Projektstand</b>			<b>Was läuft gut?</b>		
<ul style="list-style-type: none"> <li>Projekt nach Zielvorgaben abgeschlossen</li> <li>Abgabe und Upload ausstehend</li> </ul>			<ul style="list-style-type: none"> <li>Abgabe :D</li> </ul>		
			<b>Was läuft nicht gut?</b>		
			-		
<b>Geplante nächste Schritte / getroffene Massnahmen</b>					
Upload für die Teko und PDF-Versand an Experten und Sekretariat					

ProjektStatusbericht; Stefan Thöni, Josef Räber

Abbildung 70 Statusbericht Sechs

## 8.8 Anhang 3: YAML Code Automatisierungen und Konfiguration

### Inhalt des configuration.yaml:

```

/config/configuration.yaml
1  ~
2  # Loads default set of integrations. Do not remove.
3  default_config: ~
4  ~
5  # Load frontend themes from the themes folder
6  frontend: ~
7  - themes: !include_dir_merge_named themes
8  ~
9  automation: !include automations.yaml
10 script: !include scripts.yaml
11 scene: !include scenes.yaml
12 http: ~
13   - ssl_certificate: /ssl/fullchain.pem
14   - ssl_key: /ssl/privkey.pem
15 ~
16 zha: ~
17   - custom_quirks_path: /config/custom_quirks/
18 ~
19 google_assistant: ~
20   - project_id: homeassistant-10218
21   - service_account: !include SERVICE_ACCOUNT.JSON
22   - report_state: true
23   - exposed_domains: ~
24     - switch
25     - light
26 ~
27 sensor: !include_dir_merge_list sensors/
28 template: !include_dir_merge_list templates/
29 ~
30 alexa: ~
31   - smart_home: ~
32     - locale: de-DE
33     - endpoint: https://api.eu.amazonalexa.com/v3/events
34     - client_id: REDACTED
35     - client_secret: !include amazon_secret.json

```

Abbildung 71 Anhang 3 - configuration.yaml

### Inhalt des automation.yaml:

```

- id: '1696169155660'
  alias: LichtSchalter Büro
  description: "
  trigger:
  - device_id: cf7400269c480a3f88bcf5b96d13fd06
    domain: zha
    platform: device
    type: remote_button_short_press
    subtype: button_1
  condition: []
  action:

```

- type: toggle
  - device\_id: ea047ec5e5b10327695514b7514248a2
  - entity\_id: 3d971e876723e3942ecf82eb476d44b9
  - domain: light
  - mode: single
- id: '1696709206326'
  - alias: Philips Hue Dimmer RWL022
  - description: "
  - use\_blueprint:
    - path: ODickins/HueDimmer\_RWL022.yaml
    - input:
      - remote: 3d9811322ae3d1bb6c9457bc80e42a96
      - power\_light: light.bulb\_rgbcw\_6ff715
      - hue\_light: light.led
- id: '1696840352271'
  - alias: 'Automatische Lichtsteuerung '
  - description: Schlafzimmer
  - use\_blueprint:
    - path: Blackshome/sensor-light.yaml
    - input:
      - light\_brightness: 35
      - light\_transition\_on: 0
      - light\_transition\_off: 0
      - include\_sun: sun\_enabled
      - after\_time: 06:30:00
      - before\_time: '22:30:00'
      - motion\_trigger:
        - binary\_sensor.bewegungsmelder\_schlafzimmer\_motion
      - light\_switch:
        - entity\_id: light.bulb\_rgbcw\_6ff715
        - include\_light\_control:
          - use\_brightness
        - include\_time: time\_enabled
- id: '1696840611873'
  - alias: Lichtsteuerung Büro
  - description: "
  - use\_blueprint:
    - path: Blackshome/sensor-light.yaml
    - input:
      - motion\_trigger:
        - binary\_sensor.bewegungsmelder\_buro\_motion
      - light\_switch:

```
    entity_id: light.licht_light
    light_transition_on: 0
    light_transition_off: 0
    include_sun: sun_disabled
    include_time: time_enabled
    after_time: 08:00:00
    before_time: '22:00:00'
    time_delay: 15
- id: '1696841375950'
  alias: Notify Open Window
  description: Schlafzimmer
  trigger:
  - type: opened
    platform: device
    device_id: b973f53fc46a0fdf48f5261b3e0cbae1
    entity_id: e21c51af8246ca0622ce27e846237e6f
    domain: binary_sensor
  condition: []
  action:
  - service: notify.notify
    data:
      message: Schlafzimmer Fenster wurde geöffnet
      title: Fenster Offen
  mode: single
- id: '1696841440001'
  alias: Notify Open Window
  description: Büro
  trigger:
  - type: opened
    platform: device
    device_id: cb696c613090c1adca68a8ae8c3cc1b9
    entity_id: 3dc914e0441a6af4dccbe133d0eec50e
    domain: binary_sensor
  condition: []
  action:
  - service: notify.notify
    data:
      message: Büro Fenster wurde geöffnet
      title: Fenster Offen
  mode: single
- id: '1696860226229'
  alias: Test Power On Logitech Z906
```

```
description: "  
trigger: []  
condition: []  
action:  
- service: zha.issue_zigbee_cluster_command  
  data:  
    cluster_type: in  
    ieee: 18:7a:3e:ff:fe:9e:89:10  
    command_type: server  
    params:  
      code: JgBYAAABJpEUERQ1FBEUEBQRFBEUERQQFBEUERISFBEUERQ1FBEUN-  
RQRFBAUERQRFBEUEBMSFDUUNRQ1FDUUNhQ1FDUUNRQQFAAF3AABJ0gUA-  
AxMAAEoRxIADQU=  
      cluster_id: 57348  
      endpoint_id: 1  
      command: 2  
    mode: single  
- id: '1696874806703'  
  alias: Wecker  
  description: "  
  trigger:  
  - platform: time  
    at: sensor.xiaomi13_next_alarm  
  condition:  
  - condition: device  
    device_id: f97c3e710818f75bccc39af978bd25c0  
    domain: device_tracker  
    entity_id: 91bb86e5d98629002e90556cd2e17459  
    type: is_home  
  action:  
  - service: media_player.media_play_pause  
    data: {}  
    target:  
      device_id: b9be54b6afa49446322c027743806b42  
    mode: single  
- id: '1697396177851'  
  alias: Turn off all lights when leaving  
  description: "  
  trigger:  
  - platform: device  
    device_id: 3ee32839ad0c9a328e17b5d3bf7adc43  
    domain: device_tracker
```

```
entity_id: aff332fef19ed5c43b0264cebad0ee4d
type: leaves
zone: zone.home
condition: []
action:
- service: light.turn_off
  data: {}
  target:
    entity_id: light.alle_lichter
mode: single
- id: '1698015922875'
  alias: Einbruchswarung
  description: "
  trigger:
  - type: opened
    platform: device
    device_id: cb696c613090c1adca68a8ae8c3cc1b9
    entity_id: 3dc914e0441a6af4dccbe133d0eec50e
    domain: binary_sensor
  - type: opened
    platform: device
    device_id: b973f53fc46a0fdf48f5261b3e0cbae1
    entity_id: e21c51af8246ca0622ce27e846237e6f
    domain: binary_sensor
  condition:
  - condition: time
    after: '23:00:00'
    before: 05:30:00
  action:
  - service: notify.notify
    data:
      message: Fenster wurden geöffnet möglicher Einbruch
      title: Einbruchswanung
mode: single
- id: '1698017488130'
  alias: Intruder Alert
  description: "
  trigger:
  - platform: conversation
    command: Intruder Alert
  condition: []
  action:
```

```
- service: flux_led.set_custom_effect
  data:
    speed_pct: 50
    transition: jump
    colors: blue red
  target:
    entity_id: light.bulb_rgbcw_6ff715
mode: single
- id: '1698017879672'
  alias: Movie Time
  description: "
  trigger:
  - platform: conversation
    command: Movie Time
  condition: []
  action:
  - type: turn_on
    device_id: 03c94289f4aa6aad4b274ed267525654
    entity_id: 9f8ec846aef1fea03aa934fbaa0518a3
    domain: light
    brightness_pct: 30
  - type: turn_on
    device_id: 9db3780baacee980ee49b87b6d30f874
    entity_id: 981327b2517432282ac827f3e7cc2958
    domain: remote
  - type: turn_off
    device_id: bbaaae3ed394696208472d4dc30f34fe
    entity_id: 8ba24680ef083a3ed9bcc3372498908e
    domain: light
mode: single
```

**Eigenerstellter Sensor für den Solar Inverter:**

```
# SolarMax 10MT2
```

```
- platform: tcp
  scan_interval: 10
  name: "SolarMax 10MT2"
  host: 192.168.178.10
  port: 12345
  payload:
  "{FB;32;82|64:UDC;IDC;UL1;IL1;PAC;PRL;TNF;KDY;KLD;KMT;KLM;KYR;KLY;KT0;TKK;
  SYS;KHR;CAC;TYP;SWV;BDN;ADR;PIN;DDY;DMT;DYZ;TMI;THR|2289}"
```

**Dazugehöriger Auswertung via eigen erstellte Vorlage:**

```
# SolarMax 10MT2 template
```

```
- sensor:
```

```
- name: "SolarMax 10MT2 00 DC Voltage"
```

```
  device_class: voltage
```

```
  unit_of_measurement: "V"
```

```
  state_class: measurement
```

```
  state: "{{ states('sensor.solarmax_10mt2').split(';')[2] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) * 0.1 }}"
```

```
- sensor:
```

```
- name: "SolarMax 10MT2 01 DC Current"
```

```
  device_class: current
```

```
  unit_of_measurement: "A"
```

```
  state_class: measurement
```

```
  state: "{{ states('sensor.solarmax_10mt2').split(';')[3] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) * 0.01 }}"
```

```
- sensor:
```

```
- name: "SolarMax 10MT2 02 AC Voltage"
```

```
  device_class: voltage
```

```
  unit_of_measurement: "V"
```

```
  state_class: measurement
```

```
  state: "{{ states('sensor.solarmax_10mt2').split(';')[4] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) * 0.1 }}"
```

```
- sensor:
```

```
- name: "SolarMax 10MT2 03 AC Current"
```

```
  device_class: current
```

```
  unit_of_measurement: "A"
```

```
  state_class: measurement
```

```
  state: "{{ states('sensor.solarmax_10mt2').split(';')[5] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) * 0.01 }}"
```

```
- sensor:
```

```
- name: "SolarMax 10MT2 04 AC Output"
```

```
  device_class: power
```

```
  unit_of_measurement: "W"
```

```
  state_class: measurement
```

```
  state: "{{ states('sensor.solarmax_10mt2').split(';')[6] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) * 0.5 }}"
```

```
- sensor:
```

```
- name: "SolarMax 10MT2 05 AC Output (relative)"
  device_class: power_factor
  unit_of_measurement: "%"
  state_class: measurement
  state: "{{ states('sensor.solarmax_10mt2').split(';')[7] | regex_findall_index('=([0-9A-F]+)') |
int(base=16) }}"

- sensor:
  - name: "SolarMax 10MT2 06 AC Frequency"
    device_class: frequency
    unit_of_measurement: "Hz"
    state_class: measurement
    state: "{{ states('sensor.solarmax_10mt2').split(';')[8] | regex_findall_index('=([0-9A-F]+)') |
int(base=16) * 0.01 }}"

- sensor:
  - name: "SolarMax 10MT2 07 Energy (today)"
    device_class: energy
    unit_of_measurement: "kWh"
    state_class: total_increasing
    state: "{{ states('sensor.solarmax_10mt2').split(';')[9] | regex_findall_index('=([0-9A-F]+)') |
int(base=16) * 0.1 }}"

- sensor:
  - name: "SolarMax 10MT2 08 Energy (yesterday)"
    device_class: energy
    unit_of_measurement: "kWh"
    state_class: total
    state: "{{ states('sensor.solarmax_10mt2').split(';')[10] | regex_findall_index('=([0-9A-F]+)') |
int(base=16) * 0.1 }}"

- sensor:
  - name: "SolarMax 10MT2 09 Energy (this month)"
    device_class: energy
    unit_of_measurement: "kWh"
    state_class: total_increasing
    state: "{{ states('sensor.solarmax_10mt2').split(';')[11] | regex_findall_index('=([0-9A-F]+)') |
int(base=16) }}"

- sensor:
  - name: "SolarMax 10MT2 10 Energy (last month)"
    device_class: energy
```

```
    unit_of_measurement: "kWh"  
    state_class: total  
    state: "{{ states('sensor.solarmax_10mt2').split(';')[12] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) }}"
```

- sensor:

```
  - name: "SolarMax 10MT2 11 Energy (this year)"  
    device_class: energy  
    unit_of_measurement: "kWh"  
    state_class: total_increasing  
    state: "{{ states('sensor.solarmax_10mt2').split(';')[13] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) }}"
```

- sensor:

```
  - name: "SolarMax 10MT2 12 Energy (last year)"  
    device_class: energy  
    unit_of_measurement: "kWh"  
    state_class: total  
    state: "{{ states('sensor.solarmax_10mt2').split(';')[14] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) }}"
```

- sensor:

```
  - name: "SolarMax 10MT2 13 Energy (total)"  
    device_class: energy  
    unit_of_measurement: "kWh"  
    state_class: total_increasing  
    state: "{{ states('sensor.solarmax_10mt2').split(';')[15] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) }}"
```

- sensor:

```
  - name: "SolarMax 10MT2 14 Temperature (inverter)"  
    device_class: temperature  
    unit_of_measurement: "°C"  
    state_class: measurement  
    state: "{{ states('sensor.solarmax_10mt2').split(';')[16] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) }}"
```

- sensor:

```
  - name: "SolarMax 10MT2 15 Operating Mode"  
    state_class: measurement  
    state: "{{ states('sensor.solarmax_10mt2').split(';')[17] | regex_findall_index('=([0-9A-F]+)') |  
int(base=16) }}"
```

- sensor:
  - name: "SolarMax 10MT2 16 Operating Hours"
    - device\_class: duration
    - unit\_of\_measurement: "h"
    - state\_class: total
    - state: "{{ states('sensor.solarmax\_10mt2').split(';')[18] | regex\_findall\_index('=([0-9A-F]+)') | int(base=16) }}"
- sensor:
  - name: "SolarMax 10MT2 17 Start-ups"
    - state\_class: total
    - state: "{{ states('sensor.solarmax\_10mt2').split(';')[19] | regex\_findall\_index('=([0-9A-F]+)') | int(base=16) }}"
- sensor:
  - name: "SolarMax 10MT2 18 Type"
    - state\_class: measurement
    - state: "{{ states('sensor.solarmax\_10mt2').split(';')[20] | regex\_findall\_index('=([0-9A-F]+)') | int(base=16) }}"
- sensor:
  - name: "SolarMax 10MT2 19 Software Version"
    - state\_class: measurement
    - state: "{{ states('sensor.solarmax\_10mt2').split(';')[21] | regex\_findall\_index('=([0-9A-F]+)') | int(base=16) }}"
- sensor:
  - name: "SolarMax 10MT2 20 Build Number"
    - state\_class: measurement
    - state: "{{ states('sensor.solarmax\_10mt2').split(';')[22] | regex\_findall\_index('=([0-9A-F]+)') | int(base=16) }}"
- sensor:
  - name: "SolarMax 10MT2 21 Network Address"
    - state\_class: measurement
    - state: "{{ states('sensor.solarmax\_10mt2').split(';')[23] | regex\_findall\_index('=([0-9A-F]+)') | int(base=16) }}"
- sensor:
  - name: "SolarMax 10MT2 22 Capacity (installed)"
    - device\_class: power

```
unit_of_measurement: "W"  
state_class: measurement  
state: "{{ states('sensor.solarmax_10mt2').split(';')[24] | regex_findall_index('=([0-9A-F]+)' ) |  
int(base=16) * 0.5 }}"
```

- sensor:

```
- name: "SolarMax 10MT2 23 Date (day)"  
state_class: measurement  
state: "{{ states('sensor.solarmax_10mt2').split(';')[25] | regex_findall_index('=([0-9A-F]+)' ) |  
int(base=16) }}"
```

- sensor:

```
- name: "SolarMax 10MT2 24 Date (month)"  
state_class: measurement  
state: "{{ states('sensor.solarmax_10mt2').split(';')[26] | regex_findall_index('=([0-9A-F]+)' ) |  
int(base=16) }}"
```

- sensor:

```
- name: "SolarMax 10MT2 25 Date (year)"  
state_class: measurement  
state: "{{ '20' ~ states('sensor.solarmax_10mt2').split(';')[27] | regex_findall_index('=([0-9A-F]+)' ) |  
int(base=16) }}"
```

- sensor:

```
- name: "SolarMax 10MT2 26 Time (minute)"  
state_class: measurement  
state: "{{ states('sensor.solarmax_10mt2').split(';')[28] | regex_findall_index('=([0-9A-F]+)' ) |  
int(base=16) }}"
```

- sensor:

```
- name: "SolarMax 10MT2 27 Time (hour)"  
state_class: measurement  
state: "{{ states('sensor.solarmax_10mt2').split(';')[29] | regex_findall_index('=([0-9A-F]+)' ) |  
int(base=16) }}"
```

## Eigenständigkeitserklärung

Der Verfasser bestätigen mit ihrer Unterschrift, dass die vorliegende Arbeit selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als die angegebenen Hilfsmittel erstellt wurde.

Die aus fremden Quellen (einschliesslich elektronischer Quellen) direkt oder indirekt übernommenen Inhalte sind als solche kenntlich gemacht.

Die Arbeit ist in gleicher oder ähnlicher Form noch nicht vorgelegt worden.

Unterschriften:

Datum/Ort:

Pascal Steiner

Riken, 23.10.23

*Pascal Steiner*