



**Hochschule für Technik  
und Wirtschaft Berlin**

*University of Applied Sciences*

Technische Spezifikation  
im Fachübergreifendes Projekt

Sprachsteuerung eines Hauses

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

Betreuer: Prof. Dr.-Ing. Christian Müller

Ort, Datum: Berlin, 12.07.2022

## Inhaltsverzeichnis

Abbildungsverzeichnis .....	II
Tabellenverzeichnis .....	III
Verzeichnis vorhandener Dokumente .....	V
<b>1 Prozessüberblick .....</b>	<b>1</b>
1.1 Smarthome Workflow .....	1
1.2 Sprachbefehl Workflow .....	2
1.3 Update Workflow .....	3
<b>2 Technische Spezifikation SW .....</b>	<b>4</b>
2.1 Überblick Komponenten .....	4
2.2 Beschreibung der Implementierung .....	5
2.2.1 Home Assistant auf Raspberry PI installieren .....	5
2.2.2 Local Tuya .....	6
2.2.3 Node-RED und Rhasspy verbinden .....	7
2.2.4 Node-RED und Home Assistant verbinden .....	8
2.3 System Infrastruktur .....	9
<b>3 Technische Spezifikation Konstruktion .....</b>	<b>10</b>
<b>4 Modul Abhängigkeiten .....</b>	<b>11</b>

## Abbildungsverzeichnis

Abbildung 1: Der Smarthome Workflow .....	1
Abbildung 2: Der Sprachbefehl Workflow .....	2
Abbildung 3: Der Update Workflow .....	3
Abbildung 4: Komponentendiagramm .....	4
Abbildung 5: Flash via Etcher .....	5
Abbildung 6: Home Assistant Einstellung Ansicht mit Local Tuya Integration .....	6
Abbildung 7: Der Ansicht von der Verbindung zwischen Node-RED und Rhasspy .....	7
Abbildung 8: Der Ansicht von der Verbindung zwischen Node-RED und Home Assistant .....	8
Abbildung 9: System Infrastruktur .....	9
Abbildung 10: Technische Zeichnung von der Baugruppen .....	10

## Tabellenverzeichnis

Tabelle 1: Version Historie.....	IV
Tabelle 2: Relevante Dokumente .....	V
Tabelle 3: Komponente und Funktionen Verbindung .....	4
Tabelle 4: Home Assistant auf Raspberry PI Tabelle.....	5
Tabelle 5: Local Tuya Tabelle .....	6
Tabelle 6: Node-RED und Rhasspy Tabelle.....	7
Tabelle 7: Node-RED und Home Assistant Tabelle .....	8
Tabelle 8: Modul Abhängigkeiten .....	11

### Copyright

© Sprachsteuerung eines Hauses

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

### Version Historie

Tabelle 1: Version Historie

Version	Datum	Verantwortlich	Änderung
0.1	07.06.2022	Kenneth	Initiale Dokumenterstellung
0.2	08.06.2022	Alle	Überblick und Workflow hinzugefügt
0.3	12.06.2022	Alle	Technische Spezifikation SW hinzugefügt
0.4	12.06.2022	Alle	Abbildung und Tabelle hinzugefügt
0.5	13.06.2022	Alle	Erweiterungen
0.6	14.06.2022	Kenneth	Letzte Überprüfung Sprint 1
1.0	14.06.2022	Kenneth	Abgabe Sprint 1
1.1	09.07.2022	Alle	Technische Spezifikation SW bearbeitet
1.2	09.07.2022	Alle	Weitere Punkte bei der Beschreibung der Implementierung hinzugefügt
1.3	10.07.2022	Alle	Noch weitere Abbildung und Tabelle hinzugefügt
1.4	11.07.2022	Alle	Erweiterungen
1.5	12.07.2022	Alle	Letzte Überprüfung Sprint 2
<b>2.0</b>	<b>12.07.2022</b>	<b>Kenneth</b>	<b>Abgabe Sprint 2</b>

## Verzeichnis vorhandener Dokumente

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

*Tabelle 2: Relevante Dokumente*

Dokument	Autor	Datum
Lastenheft-Gruppe-4.pdf	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	26.04.2022
Projektplanung.mpp	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	12.07.2022
Pflichtenheft-Gruppe-4.pdf	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	24.05.2022
Qualitätsicherung-Gruppe-4.pdf	<ul style="list-style-type: none"><li>• Azim Izzudin Ramadhani Mubarak</li><li>• Bashar Mustafa</li><li>• Kenneth Austin</li><li>• Reynaldo Domenico</li></ul>	12.07.2022

## 1 Prozessüberblick

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

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

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

Um alle Komponenten abzudecken, verwenden wir ein Gehäuse, das mit 3D-Drucker gedruckt wird. Für die Software wird dieses Gerät, Rhasppy als Spracherkennung und Home Assistant zum Implementieren des Befehls verwendet. Um die Verbindung zwischen Geräte und Home Assistant zu erschaffen wird mit Local Tuya verwendet als der Home Assistant Integration.

### 1.1 Smarthome Workflow

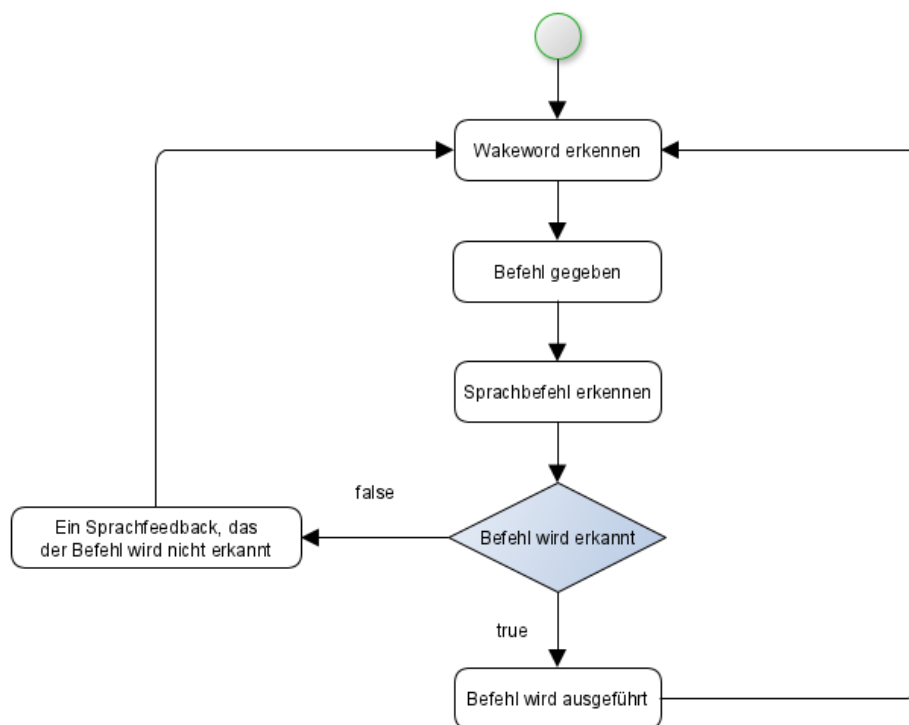


Abbildung 1: Der Smarthome Workflow

## 1.2 Sprachbefehl Workflow

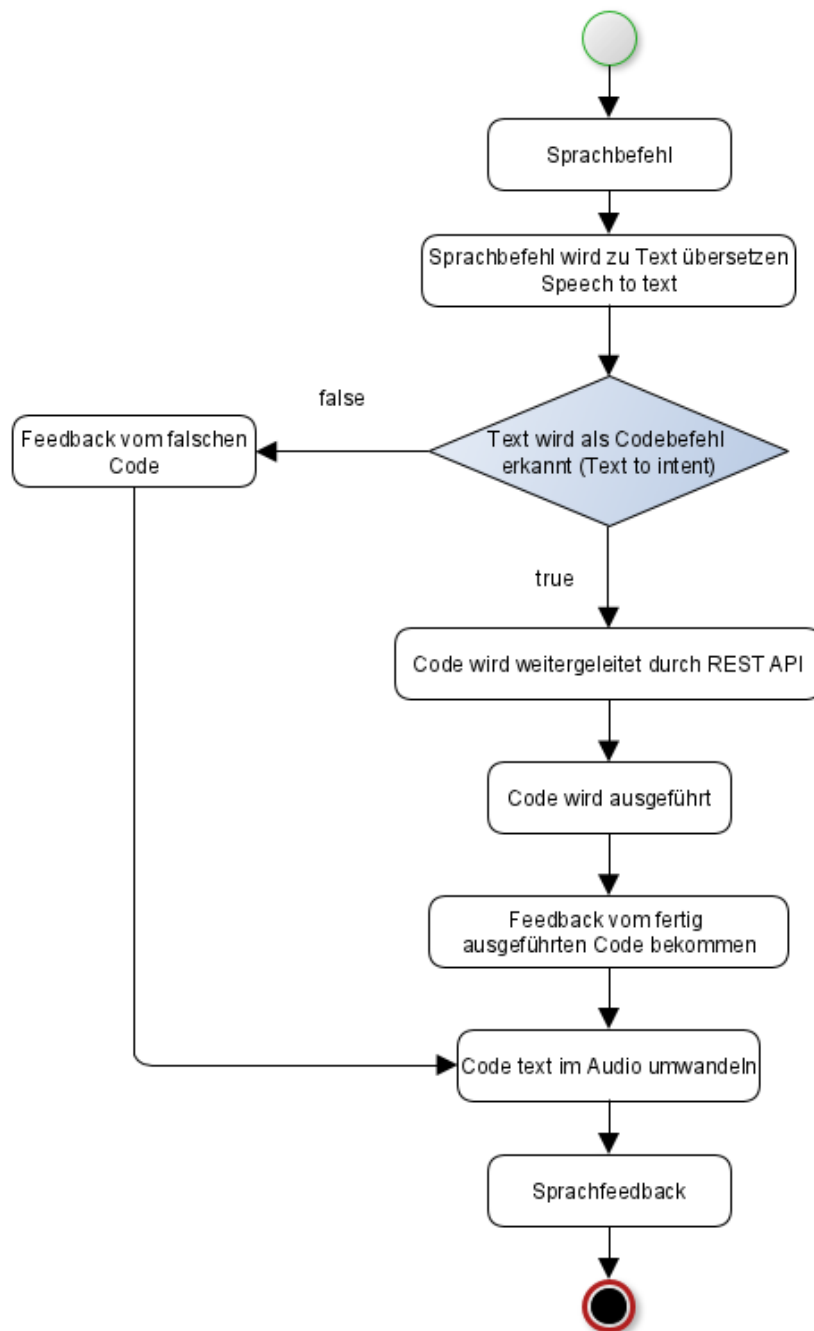


Abbildung 2: Der Sprachbefehl Workflow



### 1.3 Update Workflow

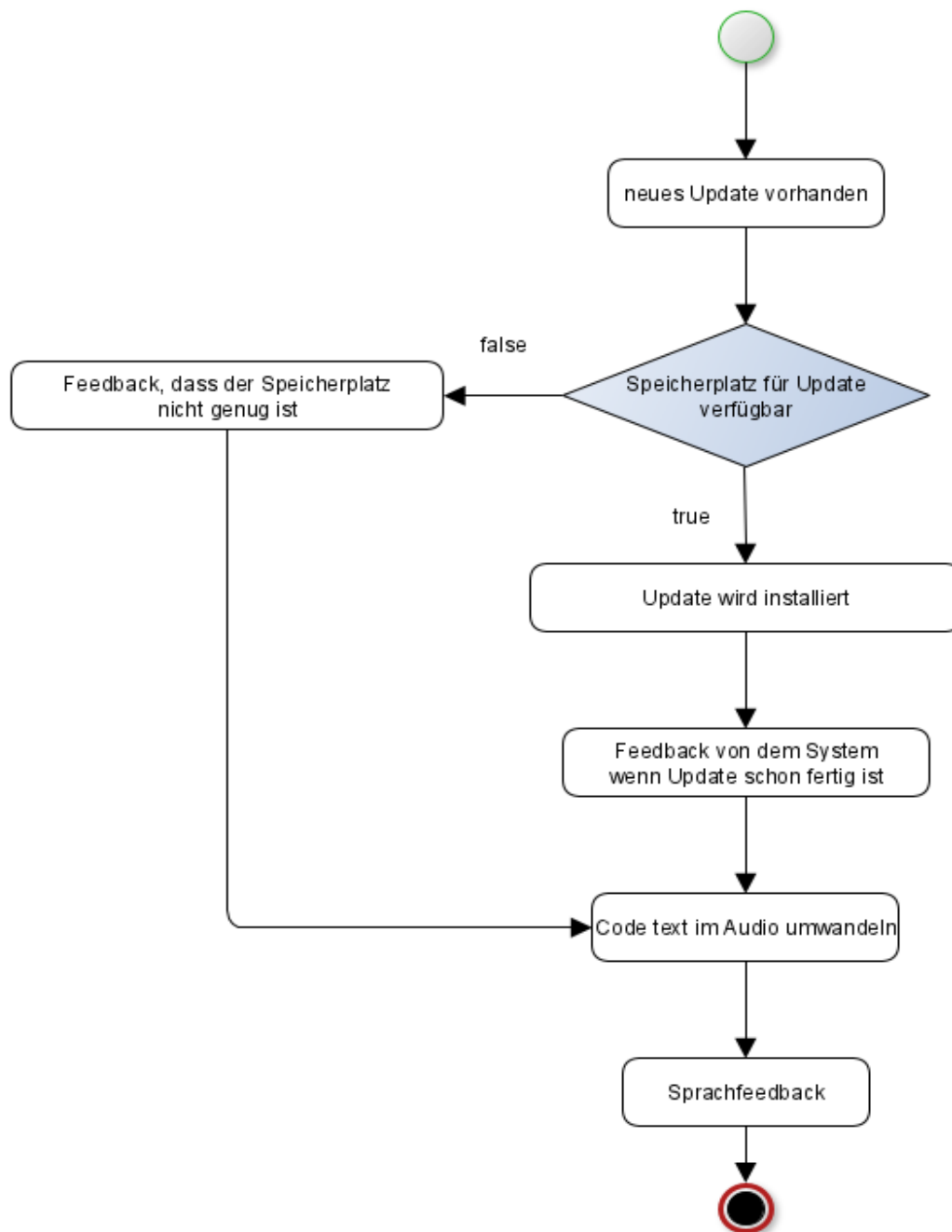


Abbildung 3: Der Update Workflow

## 2 Technische Spezifikation SW

### 2.1 Überblick Komponenten

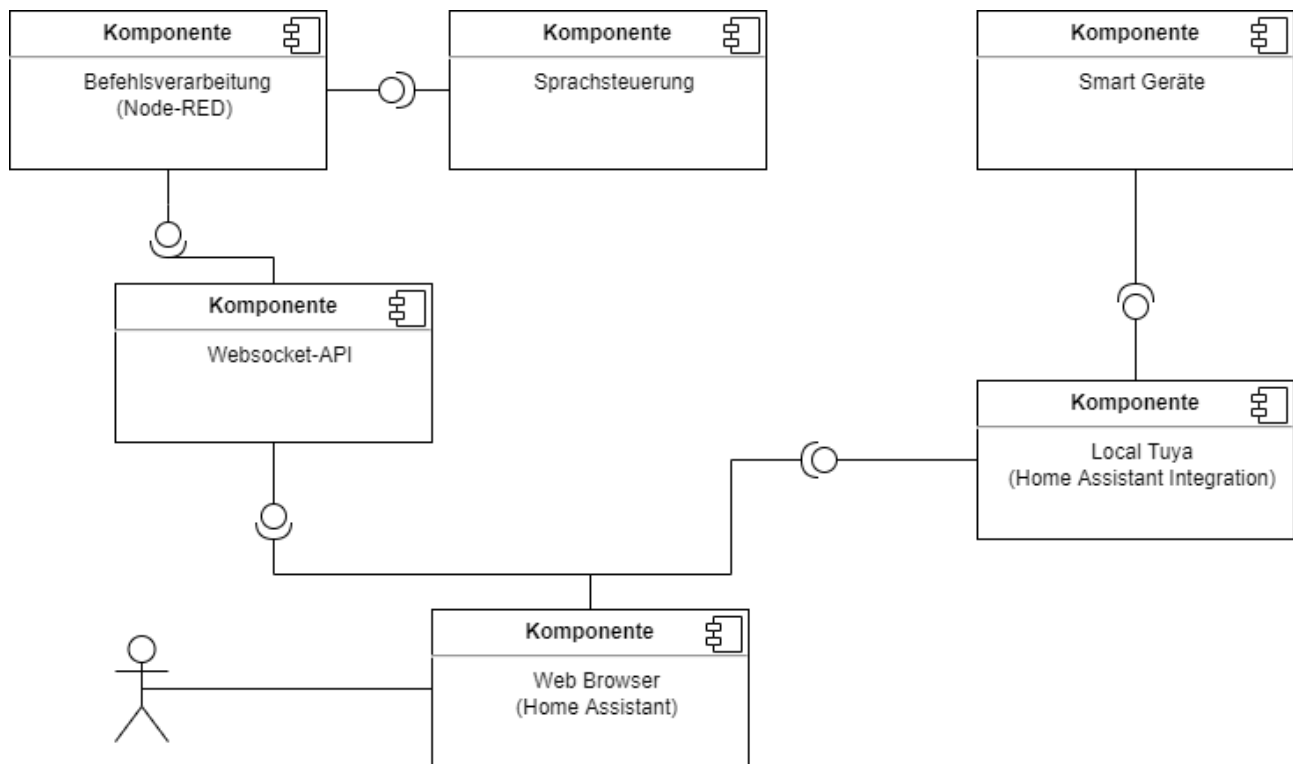


Abbildung 4: Komponentendiagramm

Tabelle 3: Komponente und Funktionen Verbindung

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

## 2.2 Beschreibung der Implementierung

### 2.2.1 Home Assistant auf Raspberry PI installieren

Dieses Arbeitspaket ist sehr wichtig, da sie von allen geplanten Funktionen benötigt wird. Hier haben wir eine Veränderung von openHAB auf Home Assistant weil die Verbindung von dem openHAB Server mit unsere Smart Geräte nicht erfolgreich verbunden deswegen haben wir Home Assistant benutzt und hat geklappt. Home Assistant wird als zentraler Steuer und auch als Verbinder zwischen Geräten und Benutzer\*in, und auch in der Sprachkontrolle benötigt.

Tabelle 4: Home Assistant auf Raspberry PI Tabelle

#	Komponentendetail	Erforderliche Arbeiten
T1	Home Assitant image	Wird verwendet, um Home Assistant auf Raspberry PI zu installieren
T2	Fernbedienung	Wird verwendet, um Raspberry PI per Home Assistant Lokal Netzwerk zu kontrollieren.

#### T1: OpenHABian image

- Image wird verwendet, um Home Assistant auf Raspberry PI zu installieren
- Image muss erstmal auf SD-Karte geflasht werden
- SD Karte wird in Raspberry Pi einsetzen und dafür muss man 15-30 Minuten warten damit man die Desktop-Fernbedienung verwenden kann

#### T2: Desktop-Fernbedienung

- Diese Funktion wird verwendet, um Raspberry Pi aus der Ferne zu kontrollieren
- Im Web Browser wird angezeigt die Verbindung zwischen Raspberry Pi und Laptop. Um es zu zeigen, wird die IP-Adresse von Raspberry PI gefordert oder direkt durch Home Assistant Lokal Netzwerk
- Lokal Netzwerk wird als die Verbindung zwischen Raspberry PI und Laptop

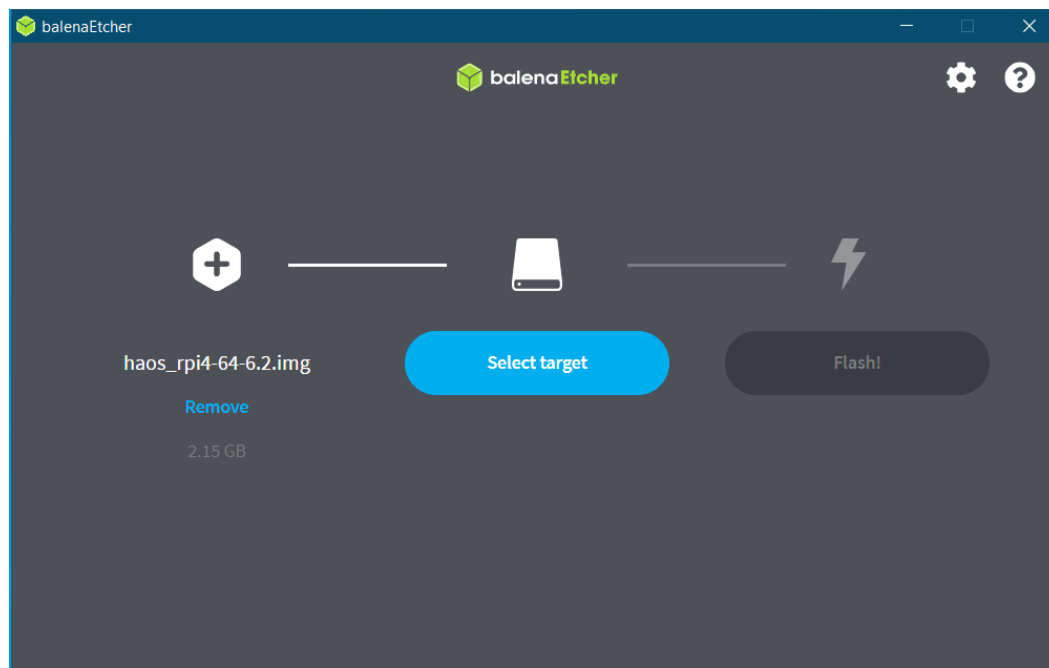


Abbildung 5:Flash via Etcher

## 2.2.2 Local Tuya

Dieses Arbeitspaket wird für die geplante Funktion benötigt, die eine Verbindung zwischen Geräte und Home Assistant sein soll. Wir haben es von MQTT auf Local Tuya umgestellt, weil mit MQTT die Verbindung nicht funktioniert und jetzt mit Local Tuya alles funktioniert hat und reibungslos läuft.

Tabelle 5: Local Tuya Tabelle

#	Komponentendetail	Erforderliche Arbeiten
T3	Home Assistant	Wird als zentraler Steuer verwendet
T4	Local Tuya	Wird als Integration zwischen unsere Smart Geräte und Home Assistant verwendet

### T3: Home Assistant

- Home Assistant ist der zentralen Steuer und auch als Verbinder zwischen Geräten und Benutzer\*in
- Home Assistant-Benutzerkonto wurde erstellt (Username="aldokece", passwort="projekt2022")
- Home Assistant bietet viele Bindungen zur Verbindung mit vielen Marken
- Bei dem Sprint 2 wird Local Tuya Integration verwendet

### T4: Local Tuya

- Local Tuya wird als Integration zwischen Home Assistant und Smart Geräte verwendet, die Tuya basiert sind.
- Vor der Verwendung von Local Tuya muss man das Lokal Key von dem Gerät wissen. Man kann durch Tuya IOT Platform das finden.

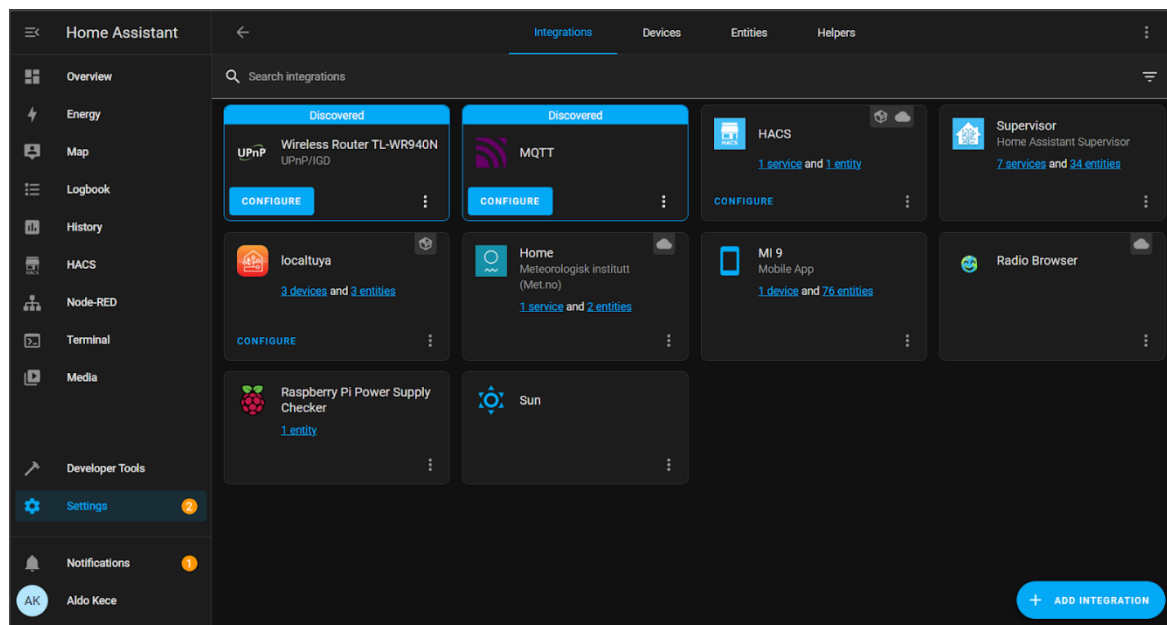


Abbildung 6: Home Assistant Einstellung Ansicht mit Local Tuya Integration

## 2.2.3 Node-RED und Rhasspy verbinden

Dieses Arbeitspaket ist nützlich, um eine Verbindung zwischen Rhasspy (Spracherkennung) und Node-RED (Befehlsverarbeitung) herzustellen. Die beiden sind über Websocket-API verbunden. Rhasspy wird den Sprachbefehl empfangen und die Bedeutung des Befehls verstehen. Diese Bedeutung (Intent) wird an Node-RED weitergeleitet und verarbeitet. Nach der Verarbeitung des Befehls gibt Node-RED Feedback, das an Rhasspy weitergeleitet wird, und später wird das Feedback in Audio-Feedback umgewandelt.

Tabelle 6: Node-RED und Rhasspy Tabelle

#	Komponentendetail	Erforderliche Arbeiten
T5	Rhasspy	Wird als Spracherkennung verwendet
T6	Node-RED	wird als zentrale Befehlsverarbeitung und Verbindung zwischen Spracherkennung und Home Assistant verwendet

### T5: Rhasspy

- Rhasspy wird als Spracherkennung verwendet
- Über Rhasspy wird die Spracheingabe und und Sprachausgabe eingestellt

### T6: Node-RED

- Node-RED wird als zentrale Befehlsverarbeitung
- Node-RED wird also als Verbindung zwischen Spracherkennung und Home Assistant verwendet
- JSON File wird verwendet, um Funktionen innerhalb Node-RED zu erstellen

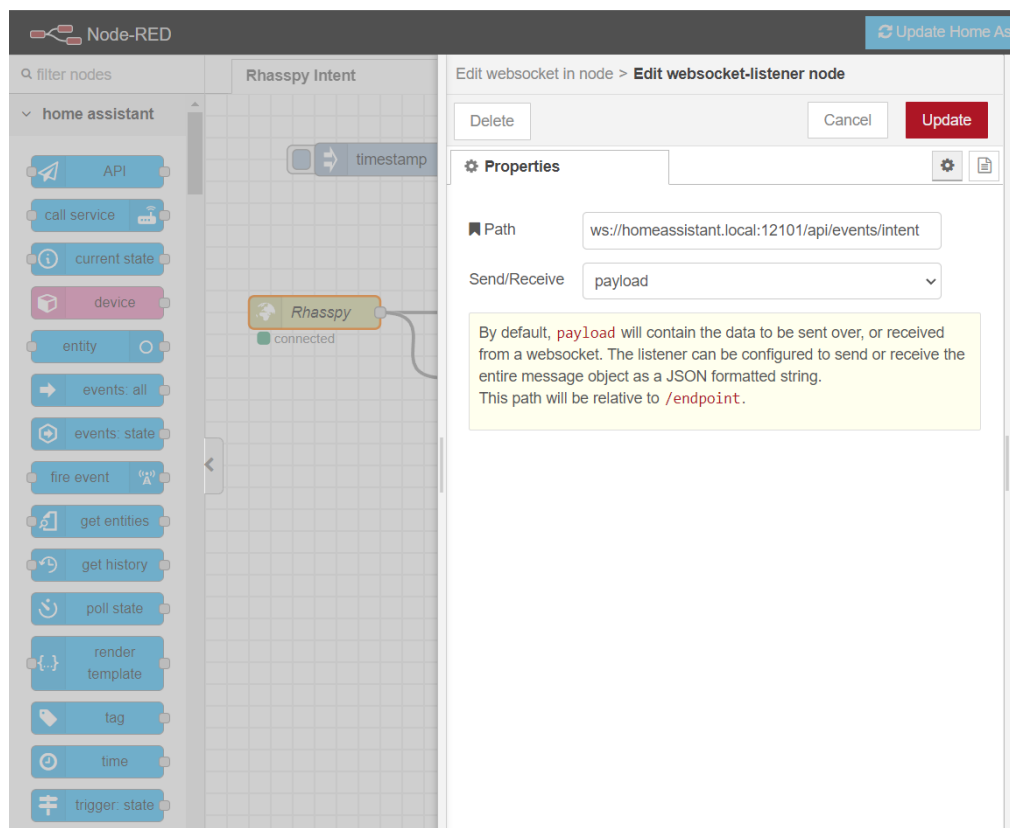


Abbildung 7: Der Ansicht von der Verbindung zwischen Node-RED und Rhasspy

## 2.2.4 Node-RED und Home Assistant verbinden

Dieses Arbeitspaket ist nützlich, um eine Verbindung zwischen Home Assistant und Node-RED (Befehlsverarbeitung) herzustellen. Die sind über Websocket-API verbunden. Node-RED wird einen Befehl von Rhasspy bekommen und wird der Befehl verarbeitet, danach wird es zum Home Assistant weitergeleitet. Es wird viele Aktionen, die man auswählen und anpassen kann nach bestimmten Befehlen.

Tabelle 7: Node-RED und Home Assistant Tabelle

#	Komponentendetail	Erforderliche Arbeiten
T3	Home Assistant	Wird als zentraler Steuer verwendet
T6	Node-RED	Wird als zentrale Befehlsverarbeitung und Verbindung zwischen Spracherkennung und Home Assistant verwendet

### T5: Home Assistant

- Home Assistant ist der zentralen Steuer und auch als Verbinder zwischen Geräten und Benutzer\*in
- Die Aktionen wird in Home Assistant ausgeführt, um die Smart Geräte zu steuern

### T6: Node-RED

- Node-RED wird als zentrale Befehlsverarbeitung
- Node-RED wird also als Verbindung zwischen Spracherkennung und Home Assistant verwendet
- JSON File wird verwendet, um Funktionen innerhalb Node-RED zu erstellen

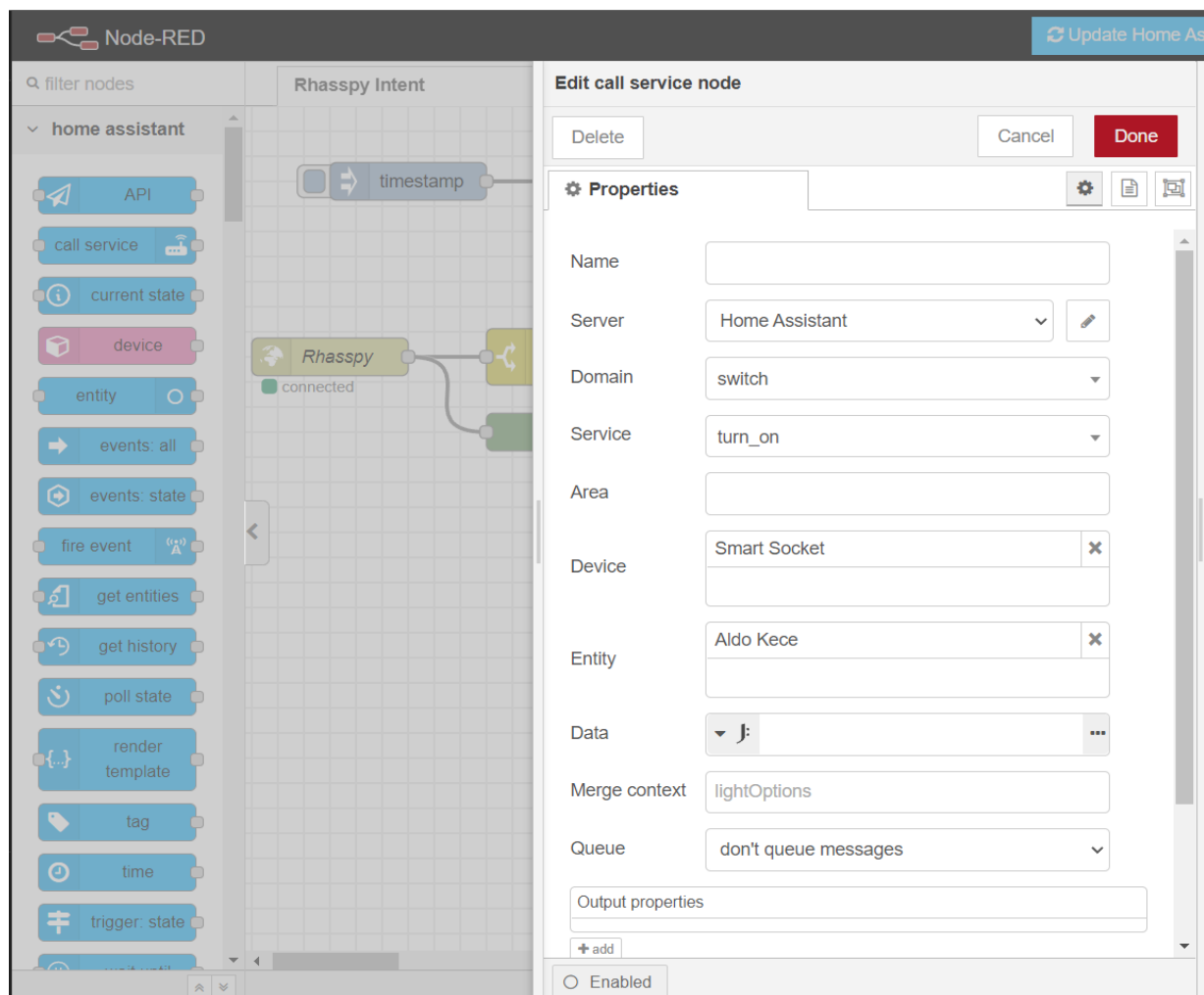


Abbildung 8: Der Ansicht von der Verbindung zwischen Node-RED und Home Assistant

## 2.3 System Infrastruktur

Die wichtigste Infrastruktur in diesem Projekt ist Home Assistant als zentraler Steuer und auch als Verbinder zwischen Geräten und Benutzer\*in. Home Assistant bietet viele Add-On und Integration zur Verbindung mit vielen Geräten. In diesem Sprint haben wir Local Tuya Integration verwendet. Node-RED und Rhasspy sind ein wichtiger Teil dieses Projekts. Node-RED wird als zentrale Befehlsverarbeitung und Verbindung zwischen Spracherkennung und Smart-Geräte verwendet, und Rhasspy wird als Spracherkennung verwendet.

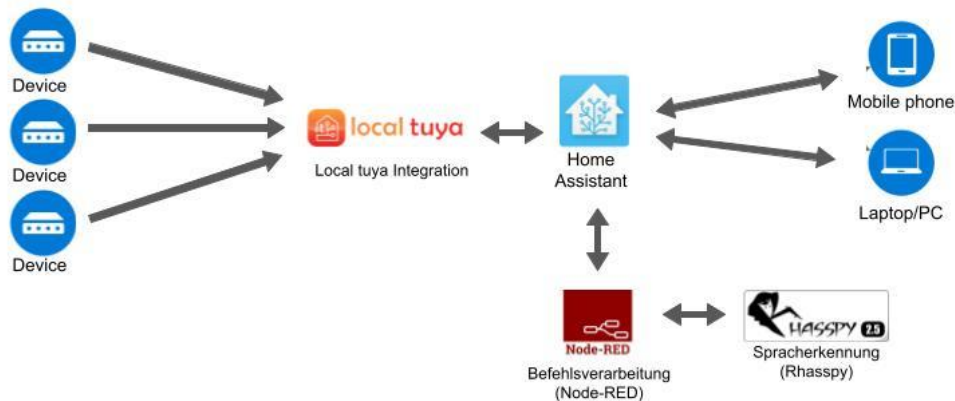


Abbildung 9: System Infrastruktur

### 3 Technische Spezifikation Konstruktion

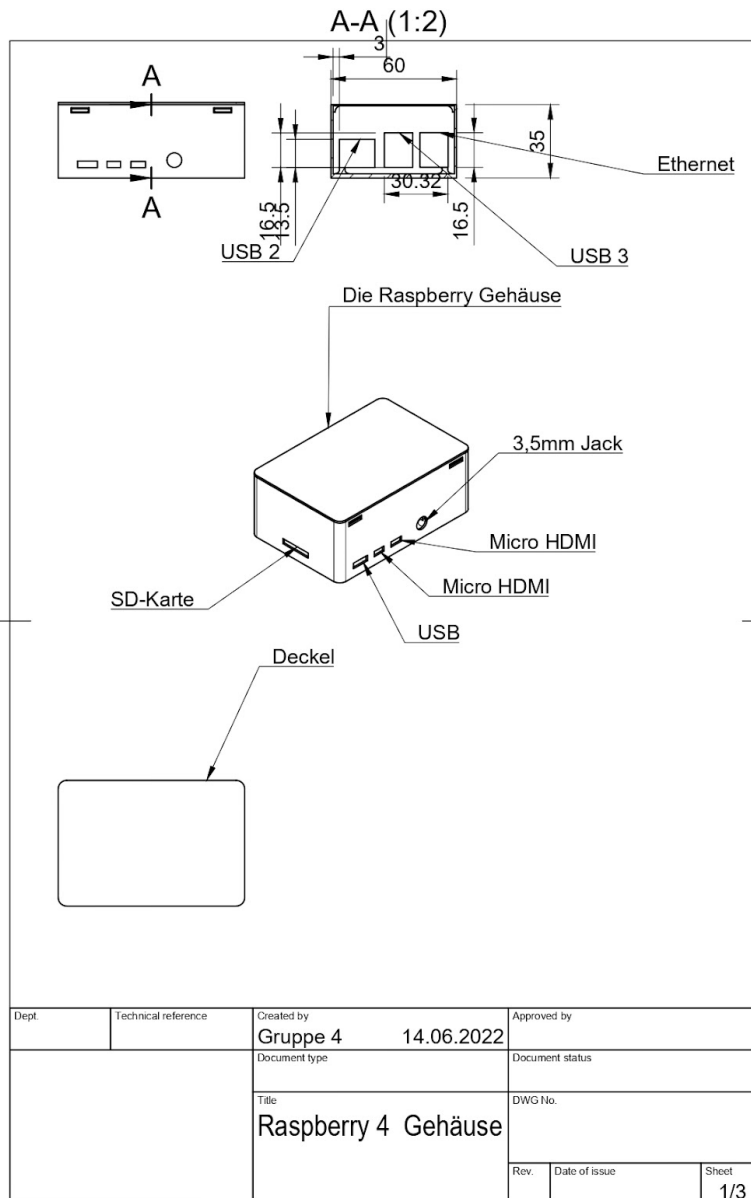


Abbildung 10: Technische Zeichnung von der Baugruppen



## 4 Modul Abhängigkeiten

Tabelle 8: Modul Abhängigkeiten

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