# Implementierung

## Einrichtung

### Raspberry Pi

Zunächst muss der Raspberry Pi eingerichtet werden [https://www.w3schools.com/nodejs/nodejs\_raspberrypi.asp]. Bei der Wahl des Geräts haben wir uns für einen Raspberry Pi 3 Model B entschieden. Im Prinzip müsste das Projekt aber auch auf alle anderen Raspberry Pi Modellen portierbar sein.

Für eine einfache Bedienung des Raspberry Pi kommt in diesem Projekt das „Raspbian Stretch with desktop“ [https://www.raspberrypi.org/downloads/raspbian/] zum Einsatz. Solange die Ansteuerung der GPIO Ports gewährleistet ist, kann auch ein beliebiges anderes Betriebssystem auf dem Raspberry Pi eingerichtet werden.

Um ein Betriebssystem für den Raspberry Pi zu installieren, muss man das Betriebssystem auf die MicroSD-Card des Raspberry Pi installiert („geflasht“) werden. Am einfachsten geht dies mit dem kostenlosen Programm Etcher [https://etcher.io/]. Etcher ist für Windows, Linux und MacOS erhältlich. In Etcher muss man lediglich das Medium (die MicroSD-Card) sowie ein Image zur Installation auswählen und die Installation starten. Sobald Etcher das Betriebssystem auf dem Raspberry Pi installiert hat, kann die MicroSD-Card in den Raspberry Pi gesteckt werden und von direkt gebootet werden.

Der Raspberry Pi kann entweder über direkt angeschlossene Peripheriegeräte bedient werden oder per Remote. Für die Bedienung über eine SSH empfiehlt sich Putty [https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html]. In unserem Projekt haben wir außerdem RealVNC [https://www.realvnc.com/de/] verwendet, welches auf dem Raspbian Stretch bereits vorinstalliert ist. RealVNC hat gegenüber SSH den Vorteil, dass der gesamte Desktop über Remote bedient werden kann.

### Python

Installierte/benötigte Python Version für elropi.py?

### Node.js

Für das Projekt wird Node.js in Version 9.x verwendet. Um das aktuellste Node.js v9.x zu installieren führt man folgende befehle aus [https://nodejs.org/en/download/package-manager/]:

curl -sL https://deb.nodesource.com/setup\_9.x | sudo -E bash -

sudo apt-get install -y nodejs

### Cloud9

[Extra? Ausführlicher?]

Für die Entwicklung auf dem Raspberry Pi haben wir die Web-DIE Cloud9 [https://github.com/c9/, https://dev.to/jtlunsford/setup-javascript-ide-on-pi-3 ] verwendet, welche auf dem Raspberry Pi als Server installiert wurde. Die Web-IDE ermöglicht besonders einfaches entwickeln und testen über jeden Webbrowser auf Computern im Selben Netzwerk wie der Raspberry Pi.

### Git

Zur Sicherung und Versionierung haben wir ein GitHub Repository eingerichtet.

git clone <https://github.com/RobinWarth/Smart-Home-Solutions.git>

git remote add origin <https://github.com/RobinWarth/Smart-Home-Solutions.git>

### Alexa Skill

Für das Erstellen von Alexa Skills muss ein Amazon Developer Konto erstellt werden [Link].

### Amazon Webservices (AWS) Lambda

Um einen Alexa Skill mit der Smart Home Skill API zu erstellen, wird ein AWS Lambda Server vorausgesetzt. Hierfür muss ein Konto für AWS erstellt und ein AWS Lambda Server gemietet werden.