

Smart-Home-Lösungen mittels Sprachsteuerung und Raspberry Pi

STUDIENARBEIT

für die Prüfung zum
Bachelor of Science
des Studienganges Angewandte Informatik
an der
Dualen Hochschule Baden-Württemberg Karlsruhe

Abgabedatum 14.05.2017

Bearbeitungszeitraum	24 Wochen
Autoren	Maximilian Hirte, Robin Warth
Matrikelnummern	8994521, 6028632
Kurs	TINF15B4
Ausbildungsfirma	Siemens AG Östl. Rheinbrückenstr. 50 76187 Karlsruhe
Betreuer der DHBW	Prof. Dr. Jürgen Röthig

Erklärung

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema: „Smart-Home-Lösungen mittels Sprachsteuerung und Raspberry Pi“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort Datum

Unterschrift

Ort Datum

Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Abkürzungsverzeichnis	VI
1 Einleitung - erster Entwurf	1
2 Aufgabenstellung	2
3 Grundlagen	3
3.1 Smart Home	3
3.2 Sprachverarbeitung	3
3.3 Analyse und Vergleich von digitalen Assistenten	5
3.3.1 Amazon-Alexa	5
3.3.2 Apple-Siri	5
3.3.3 Google-Assistent	6
3.3.4 Microsoft-Cortana	6
3.3.5 Samsung-Bixby	7
3.4 Raspberry Pi	7
3.5 Funkstandards	7
3.6 Development-Tools	7
3.6.1 Frameworks	8
3.6.2 Amazon Developer Services	8
3.7 Gebrauchstauglichkeit	8
4 Konzept	10
4.1 Anforderungsanalyse	10
4.1.1 Grund der Umsetzung	10
4.1.2 Funktionale Anforderungen	10
4.1.3 Nicht-Funktionale Anforderungen	11
4.2 Herangehensweise	12
4.3 Architektur	12
4.4 Design	12
5 Implementierung	14
6 Ergebnis, Fazit und Ausblick	15

Literatur

VIII

Anhang

X

Abbildungsverzeichnis

Abb. 1: Raspberry Pi-Alexa-Kommunikation [ABB01]	13
--	----

Abkürzungsverzeichnis

API Application Programming Interface

HTTP Hypertext Transfer Protocol

LED lichtemittierende Diode

1 Einleitung - erster Entwurf

Wir leben mittlerweile in einer Welt, in der jegliche Teilprozesse des täglichen Lebens bereits automatisiert sind bzw. automatisiert werden sollen, so auch das eigene zu Hause. Seit Anfang des Jahrtausends stellten sich Firmen und Institute die Aufgabe, das intelligente Zuhause entwickeln [VOE16] und schon mehrere Jahre ist der Begriff Smart Home in aller Munde. Es werden jegliche Systeme innerhalb des Eigenheims zentral gesteuert und mit einander verknüpft. Über eine App kann dann der Benutzer seine persönlichen Einstellungen vornehmen und das Lebensgefühl somit auf eine neue Ebene setzen.

In letzter Zeit geht der Trend dahin, diese Hausautomatisierung per Sprachbefehle zu koordinieren. Zu diesem Zweck gibt es verschiedene unabhängige digitale Sprachassistenten, die dem Bewohner das Leben vereinfachen. Besonders Amazon-Alexa und der Google-Assistent ringen im Kampf um die Marktvormachtstellung und überzeugen beide im Bereich der Sprachverständnis und -verarbeitung, aber vor allem auch mit den Fähigkeiten, die sie bieten.

Jedoch bringt eine komplette Wohneinrichtung mit Systemen, die von Haus aus die Verbindung zu den digitalen Assistenten gewährleisten, immense Kosten mit sich. Die ließen sich reduzieren, wenn sich z. B. nicht nur die von Amazon und Google vorgesehenen und zu hohen Preisen angepriesenen Geräte koppeln lassen würden, sondern jegliche Systeme, Geräte und Anlagen, die für den Smart Home-Gebrauch geeignet sind.

Die Motivation dieser Studienarbeit ist somit, eine solche Kostenersparnis zu erzielen. Amazon bietet mit seinem Developer Services die Möglichkeit, Alexa mit einem Raspberry Pi zu verbinden oder sogar direkt auf Kleincomputer zu verbinden. Die Aufgabe besteht somit darin den Raspberry Pi mit Alexa über normale Funkstandards mit Geräten zu verbinden, die ansonsten über keinerlei Fähigkeiten verfügen, sich mit digitalen Sprachassistenten zu verknüpfen. Neben der Verringerung der Kosten bietet eine solche Lösung eine weitere Stufe der Individualisierbarkeit und vor allem Geräte- bzw. Herstellerunabhängigkeit für die persönliche Smart Home-Zusammenstellung in den eigenen vier Wänden.

2 Aufgabenstellung

- Vergleich von digitalen Assistenten
- mindestens Funksteckdosen
- eigene (Überwachungs-)Kamera
- Kostenersparnis aufzeigen
- eigene Skills zur Ansteuerung

3 Grundlagen

In diesem Kapitel werden der Arbeit zu Grunde liegende Begriffe erklärt, Zusammenhänge zwischen diesen aufgezeigt sowie verschiedene Programme vorgestellt und verglichen.

3.1 Smart Home

Unter dem Begriff Smart Home wird eine Menge von technischen Systemen, Geräten und Anlagen im Haushalt verstanden, die miteinander interagieren und zentral gesteuert werden. Die damit verbundenen Ziele sind:

- Automatisierung von Alltagsvorgängen,
- Verbesserung von Wohn- und Lebensqualität,
- Erhöhung der häuslichen Sicherheit sowie
- effiziente Energienutzung. [DEV17]

Die gesamte Haushaltstechnik wird somit vernetzt und kann individuell angepasst und automatisiert werden. So können zum Beispiel Geräteeinstellungen vorgenommen werden und über eine Schnittstelle, wie Computer oder Smartphone, verwaltet werden. Es ist somit zum Beispiel möglich seine Heizung per Zeitsteuerung zu kontrollieren oder auch per App auf dem Smartphone die Heizung, das Licht, den Fernseher oder jedes andere angeschlossene Gerät an- und abzuschalten, ohne zu Hause zu sein. Prinzipiell lassen sich alle Geräte ansteuern, die über WLAN, Bluetooth, ZigBee oder andere Funkstandards verfügen. [SCH17]

Die nächste Stufe der Entwicklung ist dann die endgerätlose Steuerung. D. h., die Geräte auch zu steuern, ohne jedes mal den PC oder die App zu benutzen. Der Fokus wird immer mehr auf Gesten- und vor allem auf Sprachsteuerung gelegt, weswegen letzteres auch das Thema dieser Arbeit ist.

3.2 Sprachverarbeitung

Seit der zügigen Entwicklung der Computertechnik in der zweiten Hälfte des 19. Jahrhunderts gibt es das Problem und den Wunsch natürliche Sprache mit Hilfe einer Maschine zu erkennen und zu verarbeiten. Denn in der heutigen Zeit ist Interaktion

mit Maschinen so gut wie unumgänglich und der herkömmliche Weg über Tastatur, Maus und Touchscreen ist für den Menschen immer noch unzureichend, denn das Hauptkommunikationsmittel ist die Sprache. [RAO17, S. 1]

Die Sprachverarbeitung unterteilt sich unter anderem in Sprachsynthese, also eine symbolischen Notation in ein Sprachsignal umzusetzen, und in Spracherkennung, die in dieser Arbeit hauptsächlich beleuchtet wird. Der Schwerpunkt hierbei liegt darin ein Sprachsignal oder auch Laut in eine textliche Form umzusetzen, sodass der Computer diese erkennt. Dies ist jedoch nach wie vor ein sehr großes Problem und wird auch nicht in den nächsten beiden Jahrzehnten so gelöst werden können, dass der Computer die menschliche Sprachwahrnehmungsfähigkeit erreicht. Der Grund dafür ist die Vielfältigkeit und Komplexität der natürlich Sprache. Auf der Ebene des Wortschatzes ist der Computer schon sehr gut aufgestellt und kann die meisten Wörter erkennen und unterscheiden, doch auf Ebene der Syntaktik und Semantik hat die Maschine bisher keine Chance. Der PC ist somit noch nicht in der Lage die Fähigkeit des Menschen, fortwährend zu untersuchen und zu entscheiden, ob das Gehörte Sinn ergibt, zu simulieren. [PFI17, S. 21]

Aus diesem Grund werden für Spracherkennungsassistenten spezielle Anwendungsszenarien konzipiert. Ein solcher digitale Assistent kann somit zwar keine vollständigen Konversationen führen, aber er verfügt zum Beispiel über die Fähigkeit einen Wetterbericht aus dem Internet zu suchen, wenn er nach dem morgigen Wetter gefragt wird. Durch solche Spezialisierung auf Anwendungsfälle werden bisher gute Resultate erreicht, da es weniger Erkennungsfehler gibt. Weiterhin ist diese Lösung kompakter, da sie sowohl weniger Speicher als auch Rechenleistung benötigt. [PFI17, S. 28]

Im Jahr 2017 hat die Mozilla Foundation das Projekt Common Voice durchgeführt, welches aus 400.000 validierten, englischen Sprachaufnahmen von 200.000 Personen besteht. Über diese Aufnahmen ließen sie verschiedene Spracherkennungssoftware drüber laufen, um somit die Fehlerquote bei der automatisierten Übersetzung der Sprachaufnahmen in Text zu minimieren. Ziel war es, eine Quote unter der Zehn-Prozent-Marke zu erreichen. Das erscheint für uns Menschen relativ viel, aber laut Forschungsergebnissen, auf die sich Mozilla beruft, liegt die Fehlerquote des menschlichen Sprachverständnis bei rund 6%. [MOZ17] Aus diesen Werten lässt sich also schließen, dass vorhandene Software im Bereich der Spracherkennung schon den menschlichen Fähigkeiten annähert.

Die maschinelle Erkennung und Verarbeitung erfolgt über Sprachsignale. Die genauen Algorithmen sowie der gesamte interne technische Ablauf der digitalen Assistenten sind nicht Teil dieser Arbeit und werden nicht weiter erläutert.

3.3 Analyse und Vergleich von digitalen Assistenten

Ein digitaler Assistent (oft auch Intelligenter Persönlicher Assistent oder auch Sprachassistent) ist ein Programm, dass durch Hilfe von Sprachverarbeitung gesprochenen Anweisungen Folge leisten kann und zudem noch in menschlicher Sprache, durch eine meist weibliche Stimme, antworten kann. Die häufigsten Anwendungsfälle stellen dabei einfache Sprachsuchen im Internet oder Abarbeiten von einfachen Aufgaben, wie Schreiben und Versenden von Nachrichten oder Erstellen von Kalendereinträgen bzw. Erinnerungen.

Im Folgenden werden die zur Zeit fünf bedeutendsten Assistenten genauer untersucht und anhand folger Kriterien miteinander verglichen:

3.3.1 Amazon-Alexa

aus [BLE17, S. 81] (heise-artikel)

+ (nerd)-humor

+ erweiterbar durch Skills

+ angenehme, freundliche Stimme

- Fremdsprachenverständnis

- Spracheinkauf nicht optimal

- auf stationären Betrieb ausgelegt

Lernverhalten gut - durch Skills

Spracherkennung fast beängstigend gut

Navigationsinformationen sehr ausbaufähig, aber Lokalservice (Kino, Veranstaltungen) gut

3.3.2 Apple-Siri

aus [BLE17, S. 84] (heise-artikel)

+ Interaktion mit Apps

+ angenehme, natürliche Stimme

+ personalisierte Ansprache

- Fremdsprachenverständnis

- deutliche Wissenslücken

- zu humorlos

Genderkorrekt auch als Mann

in erster Linie Touchbedienung ersetzen

Interaktion mit Apps sehr gut -> Taschenrechner, Email, Kalender, Notizen

Wissensfragen willkürlich mündlich oder visuell, manchmal falsch (auch Inhalt)
ortsbezogene Sachen manchmal nicht ausgereift
vertrauensenerweckend, fast freundschaftlich -> personalisiert
stellt sich auf Stimme des Anwenders ein

3.3.3 Google-Assistent

aus [BLE17, S. 83] (heise-artikel)

+ gute Smartphone-Integration

+ ortsbezogene Informationen

+ kontextbezogene Nachfragen

+ Sprachübersetzung ganzer Sätze

- etwas hochnäsige, humorarme Sprachausgabe -> in Kommunikation Gefühl große Rolle, man fragt lieber den, der freundlicher ist, mit dem man lieber redet

- eingeschränkte Funktion auf Google Home

Entscheidung, ob Ergebnis mündlich oder visuell meist richtig

Vorteile durch Googles riesigen Datenfundus

im Allgemeinwissen führend

Interaktion App-übergreifend funktioniert sehr gut: Mache Foto, schreibe eine WhatsApp, sogar Flugmodus (eigenes Schließen / Beenden)

analysiert Fragen und merkt sich Informationen darin, besserer Zusammenhang zu nachfolgenden Fragen -> besser als andere Assistenten

Sprachsynthese so gut, dass man alles versteht, dennoch recht roboterhafte Stimme

3.3.4 Microsoft-Cortana

aus [BLE17, S. 82] (heise-artikel)

++ Übersetzungsfähigkeiten

+ öffnet Programme per Sprache

- eingeschränkte Sprachbefehle

- Edge und Bing-Zwang -> anschließendes Browser-Schließen nur mit Maus

- keine reinen Audio-Informationen -> viele Fragen nicht selbst und nicht Sprache > Bing-Suche

- etwas hölzerne Stimme

hohe Verbreitung durch Windows 10

Suchergebnisse oft nicht passend: Seit wann gibt es dich? -> seid-seit.de

hat Empathie und Mitgefühl in Antworten

3.3.5 Samsung-Bixby

sdmfsdan

3.4 Raspberry Pi

ydmdf

3.5 Funkstandards

ism-band von 433,05 bis 434,79 MHz

industrial, scientific, medical band

wir benutzen 433 Mhz (70-zentimeter-band)

betriebsfunk und amateur-funk, handfunkgeräte

als ism-band für z.b. funkschalter, funkthermometer, funksteckdosen

max. leistung von 10mW

reichweite 0,3 in stadt, 2,5 auf freier fläche (quasioptisch)

69 kanäle, gebührenfrei, anmeldefrei

«Non specific Short-Range-Devices» (SRD) – Geräte geringer Sendeleistung und kurzer Sendereichweite – dürfen ohne weitere Frequenzzuteilung betrieben werden, sofern sie die technischen Voraussetzungen erfüllen

geräte für anwendungen kommen sich selten in die quere, trotz große verbreitung (garagentoröffner, autoschüssel..), da aktivierungsradius gering (<100m) und die nutzungsdauer kurz (0,1 bis 3s). trotzdem kann vorkommen dass 433,92Mhz blockiert ist, wegen z.b. funkköpfhörer, der im umkreis von 50m auf derselben frequenz arbeitet
nachteile: reichweite, kosten und stromverbrauch

nutzung nicht beschränkt -> störungen durch nachbarn wahrscheinlich
sichere übertragung nur bei leistungsfähiger kanalcodierung, module mit mehreren kanäle haben deutliche vorteile, indem sie auf andere frequenz ausweichen können
(<http://rn-wissen.de/wiki/index.php?title=Funkmodule>)

3.6 Development-Tools

fhfcdfds

3.6.1 Frameworks

dfsadf

3.6.2 Amazon Developer Services

Mit den Amazon Developer Services wird jedem Benutzer, der eine Mailadresse besitzt, die mit einem Amazon-Konto verknüpft ist, die Möglichkeit geboten, eigene Applikationen für jegliche Geräte auf Android oder iOS basierend oder für die Amazon eigenen Fire Tablets und Fernseher kostenlos zu entwickeln, zu builden, zu testen und sogar über den Amazon Appstore zu vertreiben. Nach der Registrierung eines Amazon Developer Accounts stehen dem Entwickler etliche Services und APIs zur Verfügung, die direkt eingebunden und benutzt werden können. Das umschließt beispielsweise vorgefertigte Login-Formulare durch Amazon-Konten, Werbung für die eigene Applikation auf den Amazon-Fire-Geräten zu schalten oder Multi-Screen-Möglichkeiten mit zwei-Wege-Kommunikation zwischen Fire TV und der Smartphone Anwendung. Weiterhin werden verschiedenste Plugins und Extensions wie Unity, Apache Cordova oder Xamarin unterstützt. [ROU17a]

Das für dieses Projekt wichtigste Werkzeug ist das Alexa Skills Kit. Dieses erlaubt eigene Skills, also Fähigkeiten, dass Alexa bestimmte eigens konzipierte Anwendungsfälle durch bestimmte Wörter erkennt und auf diese reagieren kann, selbst zu erstellen bzw. zu programmieren. Diese selbsterstellten Skills können dann auch über den Alexa Skills Store vertrieben werden. [ROU17b]

Über die eigenen Skills können dann über die Verbindung mittels des Raspberry Pi die angeschlossenen Geräte wie Funksteckdosen oder die Kamera, welche sonst nicht von Alexa aus erreichbar wären, angesprochen werden.

3.7 Gebrauchstauglichkeit

Gemäß „DIN EN ISO 9241“ [ISO06] sind die Kriterien für die Gebrauchstauglichkeit (Usability) über die sieben „Grundsätze der Dialoggestaltung“ beschrieben:

- Aufgabenangemessenheit
 - „Ein interaktives System ist aufgabenangemessen, wenn es den Benutzer unterstützt, seine Arbeitsaufgabe zu erledigen.“
- Selbstbeschreibungsfähigkeit
 - „Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer zu jeder Zeit offensichtlich ist, in welchem Dialog, an welcher Stelle

im Dialog er sich befindet, welche Handlungen unternommen werden können und wie diese ausgeführt werden können.“

- Erwartungskonformität
 - „Ein Dialog ist erwartungskonform, wenn er den aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen sowie allgemein anerkannten Konventionen entspricht.“
- Lernförderlichkeit
 - „Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen der Nutzung des interaktiven Systems unterstützt und anleitet.“
- Steuerbarkeit
 - „Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.“
- Fehlertoleranz
 - „Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.“
- Individualisierbarkeit
 - „Ein Dialog ist individualisierbar, wenn Benutzer die Mensch- System- Interaktion und die Darstellung von Informationen ändern können, um diese an ihre individuellen Fähigkeiten und Bedürfnisse anzupassen.“

4 Konzept

In diesem Abschnitt geht es um den grundsätzlichen Entwurf und das Konzept des praktischen Teils der Studienarbeit. Hier wird dargestellt, welches konkrete Ziel mit der Studienarbeit verfolgt und mit welchen Mitteln es erreicht werden soll.

4.1 Anforderungsanalyse

In der Anforderungsanalyse wird beschrieben, welche funktionalen und welche nicht-funktionalen Anforderungen an das Projekt gestellt sind.

4.1.1 Grund der Umsetzung

[folgt]

4.1.2 Funktionale Anforderungen

Nach Beendigung des Projekts soll die Möglichkeit bestehen, Geräte per Sprachbefehle ein und aus zu schalten. Einige der Geräte sollen über Funksteckdosen, als Steuerelement, an das Smart Home angebunden werden. Hierfür stehen beliebig viele Geräte zur Verfügung, wie beispielsweise Lampen, LED-Streifen, Ventilatoren und weitere Stromverbraucher.

Zudem soll eine Kamera, angeschlossen an einen Raspberry Pi, über Sprachbefehle bedient werden können. Es sollen verschiedene Einsatzmöglichkeiten für die Kamera geschaffen werden, wie Zeitraffer-Aufnahmen, Überwachungskamera oder stationärer Fotoapparat mit Fernsteuerung über Sprachbefehl und Webinterface.

Die Funksteckdosen, die Kamera und eventuell weitere Geräte sollen neben Sprachbefehlen auch über eine webbasierte Hauszentralsteuerung erreichbar sein. Diese Hauszentralsteuerung muss von beliebigen und webbrowsersfähigen Geräten, mit Anbindung ans Heimnetz, ansteuerbar sein. Die webbasierte Hauszentralsteuerung soll zudem über ein passendes Untermenü die zur Verfügung stehenden Funktionen aufzeigen und eine Übersicht über mögliche Sprachbefehle bieten. Über die webbasierte Hauszentralsteuerung sollen die selben Funktionen vorliegen, welche auch über Sprachbefehle zur Verfügung stehen.

Zur einfacheren Ansteuerung, erleichterter Bedienbarkeit und Komfort sollen personalisierte Befehle und Befehlsprotokolle erstellt werden können. Idealerweise im Webinterface der Hauszentralsteuerung. Zu diesen Befehlen und Protokollen zählen Befehle zur parallelen Ansteuerung von verschiedenen Geräten, Zeitschaltuhren sowie Protokolle für Energiesparmaßnahmen und Sicherheit.

(- Ansteuern von Geräten mittels Infrarotsender)

4.1.3 Nicht-Funktionale Anforderungen

Systemumgebung

Der Zugang zum System erfolgt über ein internes lokales Heimnetzwerk, über Webbrowser und Sprachbefehle. Funktionsrealisierung erfolgt mittels Node.js Server auf Raspberry Pi, Alexa Voice Service unterstützendem Gerät, wie zum Beispiel Amazon Echo Dot, und mit an Raspberry Pi verbundenen Geräten.

Das Aufrufen der webbasierten Heimzentralsteuerung ist betriebssystemunabhängig und kann von beliebigen Geräten mit JavaScript fähigem Webbrowser erfolgen, wie beispielsweise Smartphone, Tablet oder Heimcomputer.

Performance

Das System soll direkt auf Sprach-Kommandos und User-Inputs mittels Webinterface reagieren. Eine maximale Verzögerung von zwei Sekunden, bis der Benutzer ein auditives oder visuelles Feedback bekommt, wird angestrebt.

Bedienbarkeit

Das System soll einfach und intuitiv bedienbar sein und daher möglichst genau den Anforderungen (Siehe 3.7 Gebrauchstauglichkeit, ISO 9241-110:2006) gerecht werden.

Das Modul soll den Benutzer unterstützen seine Arbeit zu machen und darf dabei keine unnötigen Informationen anzeigen oder dem Benutzer überflüssige Arbeitsschritte auferlegen. Es soll dem Benutzer in jedem Moment klar sein, was er tun kann und was er zu tun hat, um sein Ziel zu erreichen. Informationen wie Rückmeldungen und Beschriftungen von Bedienelementen sowie Funktionen sollen auf den Benutzer natürlich wirken und ihm allgemein bekannt vorkommen. Hinweise sollen dem Benutzer helfen die Bedienoberfläche der webbasierten Heimzentralsteuerung und der Sprachsteuerung kennenzulernen und ihre Funktionen zu verstehen. Bei falscher Bedienung sollen Fehler

vom System erkannt werden, dem Benutzer klar gemacht werden was schiefgelaufen ist und ihm somit helfen es beim nächsten Versuch besser zu machen. Der Benutzer muss immer die Kontrolle über das System haben um sein Smart Home zu bedienen und ungewünschtes Verhalten abwenden können. Zudem soll der Benutzer sein Smart Home individualisieren können, daher muss die Möglichkeit zur individuellen Benennung von Befehlen und somit auch Geräten geschaffen werden.

Zuverlässigkeit

Das System, inklusive Sprach- und Webservice, muss allzeit verfügbar sein und nach Unterbrechung, wie zum Beispiel durch einen Stromausfall schnell und möglichst autonom anlaufen. Für Aktionen wie Sprachbefehle oder weitere, durch den Benutzer durchgeführte, Aktionen besteht eine angestrebte Fehlertoleranz von unter zehn Prozent, damit das System allzeit zuverlässig seine Arbeit verrichtet.

4.2 Herangehensweise

4.3 Architektur

Das System ist unterteilt in einer Art von Client-Server-Architektur. Der Client wird hierbei repräsentiert von dem Alexa Voice Service und der Heimzentralsteuerung im Webbrowser, da diese in erster Linie dazu dienen, Kommandos vom Benutzer entgegen zu nehmen und zudem Informationen über das System bereit zu stellen.

Die Server-Seite wird vertreten vom Raspberry Pi und dem darauf laufenden Node.js Laufzeitumgebung mit Express.js Webserver. Über den Raspberry Pi werden zudem die Smart Home Geräte angesteuert und deren Logik hinterlegt.

Die Verbindung von Server und Client erfolgt über HTTP- und TCP-Protokoll.

4.4 Design

Auf Abbildung 1 kann man die Trennung und den Zusammenhang zwischen Raspberry Pi (Server) und Amazon Echo Dot (Client) erkennen.



Abb. 1: Raspberry Pi-Alexa-Kommunikation [ABB01]

Auf dem Raspberry Pi befindet sich die Node.js Laufzeitumgebung mit Express.js Webserver. Der Express.js Webserver kommuniziert über HTTP mit dem Alexa Voice Service über ein neues Alexa Skill, welches speziell für die Anbindung von Raspberry Pi geschaffen wird.

5 Implementierung

dsfsdadas

6 Ergebnis, Fazit und Ausblick

dfsgjl

Literatur

- [ABB01] R. WARTH. *Raspberry Pi-Alexa-Kommunikation*. Bilder entstammen aus: <https://www.raspberrypi.org/app/uploads/2011/10/Raspi-PGB001.png>, <https://nodejs.org/static/images/logos/nodejs-new-pantone-black.png>, https://images-eu.ssl-images-amazon.com/images/I/41iz5Tw82IL._SY300_QL70_.jpg und https://www.wink.com/img/vendor/logo_amazon.svg; abgerufen am 01.12.2017 15:23 Uhr. 2017 [siehe S. 13].
- [BLE17] Reche M. BLEICH H. „Freundlich, hölzern, clever und arrogant - Vier Sprachassistenten in ihrem natürlichen Umfeld“. In: *c't 22/2017* [2017]. Online unter <https://www.heise.de/ct/ausgabe/2017-22-Vier-Sprachassistenten-in-ihrem-natuerlichen-Umfeld-3853597.html> - abgerufen am 13.11.2017 10:58 Uhr, S. 80–85 [siehe S. 5, 6].
- [DEV17] DEVOLO AG. *Definition Smart Home*. abgerufen am 17.11.2017 14:22 Uhr. 2017. URL: <https://www.devollo.de/smart-home/> [siehe S. 3].
- [DIS17] F. DISSELHOFF. *Google Home oder Amazon Echo? Der Vergleichstest [mit Video]*. abgerufen am 09.11.2017 12:14 Uhr. 2017. URL: <https://curved.de/reviews/google-home-oder-amazon-echo-der-vergleichstest-mit-video-516362>.
- [DON14] W. DONAT. *Learn Raspberry Pi - Programming with Python. learn to program on the world's most popular tiny computer*. USA: apress, 2014. ISBN: 978-1-4302-6425-5.
- [HEI12] H. HEIKO. *elropi.py - Remote switch Elro using Python on Raspberry PI*. abgerufen am 10.11.2017 14:01 Uhr. 2012. URL: <https://pastebin.com/aRipYrZ6>.
- [HET17] M. L. HETLAND. *Beginning Python. From Novice to Professional*. 3. Aufl. Trondheim, Norwegen: apress, 2017. ISBN: 978-1-4842-0028-5.
- [ISO06] *ISO 9241: Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung*. Standard. 2006 [siehe S. 8].

- [MER16] N. MERCIER. *Run Python scripts from Node.js with simple (but efficient) inter-process communication through stdio*. abgerufen am 14.11.2017 10:54 Uhr. 2016. URL: <https://github.com/extrabacon/python-shell>.
- [MOZ17] H. BRAUN. *Mozilla Common Voice: Sprachsteuerung für alle und ohne Rückgriff auf die Cloud*. abgerufen am 06.12.2017 14:01 Uhr. 2017. URL: <https://www.heise.de/newsticker/meldung/Mozilla-Common-Voice-Sprachsteuerung-fuer-alle-und-ohne-Rueckgriff-auf-die-Cloud-3904454.html> [siehe S. 4].
- [NJSoJ] NODE.JS Foundation. *Node.js Tutorial*. abgerufen am 14.11.2017 10:02 Uhr. o.J. URL: <https://nodejs.org/en/docs/>.
- [PFI17] Kaufmann T. PFISTER B. *Sprachverarbeitung. Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. 2. Aufl. Zürich, Schweiz: Springer Vieweg, 2017. ISBN: 978-3-662-52838-9 [siehe S. 4].
- [PSF17] Python Software FOUNDATION. *Python 3.6.3 documentation*. abgerufen am 14.11.2017 18:07 Uhr. 2017. URL: <https://docs.python.org/3/>.
- [RAO17] Manjunath K.E. RAO S. K. *Speech Recognition Using Articulatory and Excitation Source Feature*. *Speech Technology*. Indien: Springer, 2017. ISBN: 978-1-4842-2922-4 [siehe S. 4].
- [ROT17] K. ROTHER. *Pro Python - Best Practices. Debugging, Testing and Maintenance*. Berlin, Deutschland: apress, 2017. ISBN: 978-1-4842-2241-6.
- [ROU17a] M. ROUSE. *Amazon Developer Services*. abgerufen am 09.01.2018 09:57 Uhr. 2017. URL: <http://searchaws.techtarget.com/definition/Amazon-Developer-Services> [siehe S. 8].
- [ROU17b] M. ROUSE. *Alexa Skills Kit*. abgerufen am 09.01.2018 10:31 Uhr. 2017. URL: <http://searchaws.techtarget.com/definition/Alexa-Skills-Kit> [siehe S. 8].
- [SCH17] K. SCHILLER. *Was ist ein Smart Home? Geräte und Systeme*. abgerufen am 10.11.2017 12:20 Uhr. 2017. URL: <https://www.homeandsmart.de/was-ist-ein-smart-home> [siehe S. 3].
- [SLIoJ] Inc STRONGLOOP. *Express - Schnelles, offenes, unkompliziertes Web-Framework für Node.js*. abgerufen am 14.11.2017 10:05 Uhr. o.J. URL: <http://expressjs.com/de/>.

- [SOF16] o.V. *How to call Python function from NodeJS*. abgerufen am 14.11.2017 19:27 Uhr. 2016. URL: <https://stackoverflow.com/questions/23450534/how-to-call-python-function-from-nodejs>.
- [SOP17] M. E. SOPER. *Expanding Your Raspberry Pi. Storage, printing, peripherals, and network connections for your Raspberry Pi*. Indianapolis, USA: apress, 2017. ISBN: 978-1-4842-2922-4.
- [VOE16] F. VÖLKELE. *Die Historie des Smart Home von 1963 - Heute*. abgerufen am 02.02.2018 10:53 Uhr. 2016. URL: https://www.smartest-home.com/blog/smart_home_historie_1939_bis_2017_frank_voelkel/ [siehe S. 1].
- [W3SCHa] o.V. *Node.js and Raspberry Pi*. abgerufen am 14.11.2017 10:00 Uhr. o.J. a. URL: https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp.
- [W3SCHb] o.V. *Node.js Tutorial*. abgerufen am 14.11.2017 10:01 Uhr. o.J. b. URL: <https://www.w3schools.com/nodejs/>.

Anhang

dsfdgs