

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 07.05.2017

| | |
|-------------------------------|--|
| Bearbeitungszeitraum | 24 Wochen |
| Autoren | Maximilian Hirte, Robin Warth |
| Matrikelnummern | 8994521, 6028632 |
| Kurs | TINF15B4 |
| Ausbildungsfirma | Siemens AG Östl. Rheinbrückenstr. 50 76187 Karlsruhe |
| Gutachter der Studienakademie | 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

Sperrvermerk

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung der Ausbildungsstätte vorliegt.

Inhaltsverzeichnis

| | |
|---|-------------|
| Abbildungsverzeichnis | V |
| Abkürzungsverzeichnis | VI |
| 1 Einleitung | 6 |
| 2 Grundlagen | 7 |
| 2.1 Smart-Home | 7 |
| 2.2 Sprachverarbeitung | 7 |
| 2.3 Analyse und Vergleich von digitalen Assistenten | 9 |
| 2.3.1 Amazon-Alexa | 9 |
| 2.3.2 Apple-Siri | 9 |
| 2.3.3 Google-Assistent | 9 |
| 2.3.4 Microsoft-Cortana | 9 |
| 2.3.5 Samsung Bixby | 9 |
| 2.4 Raspberry Pi | 9 |
| 2.5 Funkstandards | 9 |
| 2.6 Development-Tools | 9 |
| 2.6.1 Frameworks | 10 |
| 2.6.2 Amazon Developer Portal | 10 |
| 2.7 Gebrauchstauglichkeit | 10 |
| 3 Konzept | 12 |
| 3.1 Anforderungsanalyse | 12 |
| 3.1.1 Grund der Umsetzung | 12 |
| 3.1.2 Funktionale Anforderungen | 12 |
| 3.1.3 Nicht-Funktionale Anforderungen | 13 |
| 3.2 Herangehensweise | 14 |
| 3.3 Architektur | 14 |
| 3.4 Design | 15 |
| 4 Implementierung | 16 |
| 5 Ergebnis, Fazit und Ausblick | 17 |
| Literatur | XIII |

Anhang

XIV

Abbildungsverzeichnis

| | |
|--|----|
| Abb. 1: Raspberry Pi-Alexa-Kommunikation | 15 |
|--|----|

Abkürzungsverzeichnis

dsgdsgfdgfdsgfd

1 Einleitung

hier kommt die einleitung ...

2 Grundlagen

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

2.1 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.¹

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.²

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.

2.2 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

¹Vgl. [DEVOLO AG 2017].

²Vgl. [SCHILLER 2017].

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.³

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 natürlichen 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.⁴

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.⁵

In diesem Jahr 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%.⁶ Aus diesen Werten kann man also schließen, dass vorhandene Software im Bereich der Spracherkennung schon den menschlichen Fähigkeiten annähert.

³Vgl. [RAO S. K. 2017] S.1.

⁴Vgl. [PFISTER B. 2017] S.21.

⁵Vgl. [PFISTER B. 2017] S.28.

⁶Vgl. [BRAUN 2017].

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.

2.3 Analyse und Vergleich von digitalen Assistenten

dsff

2.3.1 Amazon-Alexa

dfdsgv

2.3.2 Apple-Siri

dfsdaf

2.3.3 Google-Assistent

sfhadf

2.3.4 Microsoft-Cortana

dasfdsg

2.3.5 Samsung Bixby

sdmfsdan

2.4 Raspberry Pi

ydmdf

2.5 Funkstandards

dfsd

2.6 Development-Tools

fhfcdfds

2.6.1 Frameworks

dfsadf

2.6.2 Amazon Developer Portal

dvsddf

2.7 Gebrauchstauglichkeit

Gemäß „DIN EN ISO 9241“⁷ 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

⁷Vgl. [ISO 9241: Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung 2006].

- „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.“

3 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.

3.1 Anforderungsanalyse

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

3.1.1 Grund der Umsetzung

[folgt]

3.1.2 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)

3.1.3 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 2.7 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.

3.2 Herangehensweise

3.3 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.

3.4 Design



Abb. 1: Raspberry Pi-Alexa-Kommunikation

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

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 die Anbindung von Raspberry Pi geschaffen wird.

4 Implementierung

dsfsdadas

5 Ergebnis, Fazit und Ausblick

sdfargdg¹ dsfsdafsd² dsfsdafsd³ dsfsdafsd⁴ dsfsdafsd⁵ dsfsdafsd⁶ dsfsdafsd⁷ dsfsdafsd⁸
dsfsdafsd⁹ dsfsdafsd¹⁰ dsfsdafsd¹¹ dsfsdafsd¹² dsfsdafsd¹³ dsfsdafsd¹⁴ dsfsdafsd¹⁵ dsfs-
dafsd¹⁶

¹Vgl. [DISSELHOFF 2017], S. 237.

²Vgl. [BLEICH H. 2017], S.82.

³Vgl. [HEIKO 2012].

⁴Vgl. [O.V. o.J. a].

⁵Vgl. [O.V. o.J. b].

⁶Vgl. [NODE.JS *Foundation* o.J.]

⁷Vgl. [STRONGLOOP o.J.]

⁸Vgl. [FOUNDATION 2017].

⁹Vgl. [O.V. 2016].

¹⁰Vgl. [MERCIER 2016].

¹¹Vgl. [PFISTER B. 2017].

¹²Vgl. [DONAT 2014].

¹³Vgl. [HETLAND 2017].

¹⁴Vgl. [ROTHER 2017].

¹⁵Vgl. [SOPER 2017].

¹⁶Vgl. [RAO S. K. 2017].

Literatur

- BLEICH H., Reche M. [2017]. „Freundlich, hölzern, clever und arrogant - Vier Sprachassistenten in ihrem natürlichen Umfeld“. In: *c't 22/2017*. 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. 17].
- BRAUN, H. [2017]. *Mozilla Common Voice: Sprachsteuerung für alle und ohne Rückgriff auf die Cloud*. abgerufen am 06.12.2017 14:01 Uhr. URL: <https://www.heise.de/newsticker/meldung/Mozilla-Common-Voice-Sprachsteuerung-fuer-alle-und-ohne-Rueckgriff-auf-die-Cloud-3904454.html> [siehe S. 8].
- DEVOLO AG [2017]. *Definition Smart Home*. abgerufen am 17.11.2017 14:22 Uhr. URL: <https://www.devollo.de/smart-home/> [siehe S. 7].
- DISSELHOFF, F. [2017]. *Google Home oder Amazon Echo? Der Vergleichstest [mit Video]*. abgerufen am 09.11.2017 12:14 Uhr. URL: <https://curved.de/reviews/google-home-oder-amazon-echo-der-vergleichstest-mit-video-516362> [siehe S. 17].
- DONAT, W. [2014]. *Learn Raspberry Pi - Programming with Python. learn to program on the world's most popular tiny computer*. USA: apress. ISBN: 978-1-4302-6425-5 [siehe S. 17].
- FOUNDATION, Python Software [2017]. *Python 3.6.3 documentation*. abgerufen am 14.11.2017 18:07 Uhr. URL: <https://docs.python.org/3/> [siehe S. 17].
- HEIKO, H. [2012]. *elropi.py - Remote switch Elro using Python on Raspberry PI*. abgerufen am 10.11.2017 14:01 Uhr. URL: <https://pastebin.com/aRipYrZ6> [siehe S. 17].
- HETLAND, M. L. [2017]. *Beginning Python. From Novice to Professional*. 3. Aufl. Trondheim, Norwegen: apress. ISBN: 978-1-4842-0028-5 [siehe S. 17].
- ISO 9241: *Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung* [2006]. Standard [siehe S. 10].
- MERCIER, N. [2016]. *Run Python scripts from Node.js with simple (but efficient) inter-process communication through stdio*. abgerufen am 14.11.2017 10:54 Uhr. URL: <https://github.com/extrabacon/python-shell> [siehe S. 17].

- NODE.JS *Foundation* [o.J.] *Node.js Tutorial*. abgerufen am 14.11.2017 10:02 Uhr. URL: <https://nodejs.org/en/docs/> [siehe S. 17].
- O.V. [2016]. *How to call Python function from NodeJS*. abgerufen am 14.11.2017 19:27 Uhr. URL: <https://stackoverflow.com/questions/23450534/how-to-call-python-function-from-nodejs> [siehe S. 17].
- [o.J. a]. *Node.js and Raspberry Pi*. abgerufen am 14.11.2017 10:00 Uhr. URL: https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp [siehe S. 17].
- [o.J. b]. *Node.js Tutorial*. abgerufen am 14.11.2017 10:01 Uhr. URL: <https://www.w3schools.com/nodejs/> [siehe S. 17].
- PFISTER B., Kaufmann T. [2017]. *Sprachverarbeitung. Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. 2. Aufl. Zürich, Schweiz: Springer Vieweg. ISBN: 978-3-662-52838-9 [siehe S. 8, 17].
- RAO S. K., Manjunath K.E. [2017]. *Speech Recognition Using Articulatory and Excitation Source Feature. Speech Technology*. Indien: Springer. ISBN: 978-1-4842-2922-4 [siehe S. 8, 17].
- ROTHER, K. [2017]. *Pro Python - Best Practices. Debugging, Testing and Maintenance*. Berlin, Deutschland: apress. ISBN: 978-1-4842-2241-6 [siehe S. 17].
- SCHILLER, K. [2017]. *Was ist ein Smart Home? Geräte und Systeme*. abgerufen am 10.11.2017 12:20 Uhr. URL: <https://www.homeandsmart.de/was-ist-ein-smart-home> [siehe S. 7].
- SOPER, M. E. [2017]. *Expanding Your Raspberry Pi. Storage, printing, peripherals, and network connections for your Raspberry Pi*. Indianapolis, USA: apress. ISBN: 978-1-4842-2922-4 [siehe S. 17].
- STRONGLOOP, Inc [o.J.] *Express - Schnelles, offenes, unkompliziertes Web-Framework für Node.js*. abgerufen am 14.11.2017 10:05 Uhr. URL: <http://expressjs.com/de/> [siehe S. 17].

Anhang

dsfdgs