Inhaltsverzeichnis

1 Einleitung

In den letzten Jahren konnten aufgrund technologischer Entwicklungen immer mehr Geräte entwickelt werden, die in den verschiedensten Bereichen des täglichen Lebens eingesetzt werden. Dabei sind die Geräte meist leistungsfähiger oder kleiner geworden, sodass sie in unterschiedlichsten Bereichen eingesetzt werden können. Ein Begriff der in den letzten Jahren für dieses Phänomen immer häufiger genutzt wurde, ist das "Internet of Things (IoT)" zu deutsch "Das Internet der Dinge". Mit dem IoT ist gemeint, dass immer mehr Geräte mit dem Internet verbunden sind und Daten austauschen. Im Rahmen dieser Entwicklung sind verschiedenste Nutzungsszenarien umgesetzt worden, die das tägliche Leben erleichtern oder automatisieren sollen. Diese Studienarbeit wird sich im folgenden mit einem speziellen Szenario befassen, welches unter anderem durch den Onlinehändler Amazon umgesetzt wurde.

Das Szenario umfasst sogenannte "Dash' Buttons". Diese Buttons sind kleine Geräte, welche an unterschiedlichen Stellen in der Wohnung des Kundens angebracht werden können und über einen Druckknopf verfügen. Nach einer Konfiguration durch den Benutzer kann mithilfe eines Drucks auf den Sensor ein zuvor ausgewähltes Produkt bestellt werden. Ein Beispiel wäre das nachbestellen von Waschpulver. So könnte der Button an der Waschmaschine befestigt sein und bei einem geringen Vorrat an Waschpulver wird der Button betätigt und das ausgewählte Waschpulver innerhalb der nächsten Tage geliefert. So entfällt für den Nutzer der händische Bestellvorgang, da dieser automatisch durch den Button und Amazon durchgeführt wird. Dieses Beispiel lässt sich natürlich auf verschiedene andere Möglichkeiten übertragen, so könnten auch Dinge, wie Kaffee. Zahnpasta, Tierfutter, Getränke oder ähnliches nachbestellt werden. Dem Nutzer wird eine Vereinfachung des Kaufprozesses versprochen und der Händler bindet den Kunden stärker an sich, da der Button direkt bei Amazon bestellt. Die Studienarbeit setzt sich zum Ziel eine Untersuchung des Amazonproduktes durchzuführen und eine Alternative zu entwickeln.

2 Aufgabenstellung

Wie bereits in der Einleitung erwähnt, soll sich diese Studienarbeit mit der Untersuchung des Amazon Dash Buttons beschäftigen und eine Alternative entwickeln. Da bereits während der Recherche vor Projektbeginn herausgefunden werden konnte, dass der Button in der Standardkonfiguration nur mit den Services von Amazon zusammenarbeiten kann, soll eine offenere Lösung entwickelt werden. Mit einer offeneren Lösung wird im Rahmen dieser Studienarbeit eine Möglichkeit definiert, die dem Kunden einen ähnlichen Funktionsumfang anbietet. Ein grundlegender Unterschied ist jedoch im Bestellprozess vorhanden. Während die Lösung von Amazon eine direkte Bestellung bei Amazon auslöst, soll die offenere Lösung eine Art Einkaufszettel bereithalten.

Der Nutzer soll dann über ein Webfrontend die Möglichkeit haben den Einkaufszettel zu betrachten und dann die entsprechenden Produkte bei seinem bevorzugten Händler zu bestellen. So kann es auch möglich sein, dass auch Produkte geführt werden, die der Nutzer nicht online bestellen kann. Der Vorteil bei dieser Lösung liegt darin, dass der Nutzer nicht an einen Händler und dessen Preise gebunden ist, sondern die Preise betrachten kann und seine Bestellung dementsprechend aufgeben kann. Im Rahmen dieses Ziel müssen verschiedene Komponenten entwickelt werden, die im Rahmen der Lösung zusammenarbeiten. Diese Komponenten lassen sich unter folgenden Kategorien zusammenfassen:

- Entwicklung von Buttons
- Entwicklung eines Frontends zur Nutzerinteraktion
- Entwicklung eines Backends zur Verarbeitung von Frontendeingaben und der Eingabe von Buttons

Die Entwicklung von Buttons lässt sich dabei in zwei Teile aufteilen. Zum einem soll der Amazon Dash Button betrachtet und analysiert werden. In diesem Rahmen soll sowohl die Einrichtung und Konfiguration als auch die Kommunikation untersucht werden. In Folge dessen soll weiterhin geprüft werden, ob der Button auch für die offene Lösung genutzt werden kann und dabei kein Produkt bei Amazon bestellt. Weiterhin sollen mithilfe von Mikrokontrollern eigene Buttons entwickelt werden, die ebenfalls ein Signal an den entsprechenden Empfänger absenden können. Bei den Buttons steht die Funktionalität im Vordergrund und die Prüfung der Machbarkeit des Projektes.

Weiterhin muss ein Frontend entwickelt werden, welches die Nutzerinteraktion ermöglicht. Das Frontend soll zur Übersicht der Einkaufsliste genutzt werden, aber auch die Verwaltung der Buttons und andere anfallende Verwaltungsfunktionen sollen ermöglicht werden. Um eine Kommunikation zwischen Frontend und den Buttons zu gewährleisten, muss ein Backend entwickelt werden, welches über Schnittstellen die Kommunikation gewährleistet und die Eingaben entsprechend verarbeitet. Als Ergebnis des Projektes soll ein eigener Button entstanden sein, der mithilfe des Backends und Frontends dem Nutzer ermöglicht diesen Button auch zu nutzen. Weiterhin soll zumindest der Amazon Dash Button untersucht worden sein und wenn möglich miteingebunden. Abschließend soll ein Fazit gezogen werden, inwiefern das Projekt umsetzbar ist.

3 Theorie

Im Rahmen dieser Arbeit wurden verschiedene Technologien und Prinzipien eingesetzt. Auf diese soll in den folgenden Unterkapiteln eingegangen werden, damit eine theoretische Grundlage vorhanden ist.

3.1 UDP

UDP steht für "User-Datagram-Protocol" und ist ein verbindungsloses Transportprotokoll. Im Open Systems Interconnection (OSI)-7-Schichten Modell arbeitet es auf der Transportebene und ist für die Zustellung von Netzwerkpaketen von einem Sender zu einem Empfänger zuständig. Im Vergleich zum Transportprotokoll TCP, welches verbindungsorientiert arbeitet, ist es wesentlich einfacher zu verarbeiten, da beispielsweise der Header wesentlich kleiner ist. Allgemein ist es sehr minimal gehalten und dadurch sehr einfach zu implementieren und für sehr einfache Anwendungszwecke geeignet. Allerdings ist auch zu erwähnen, dass es keine Empfangsbestätigung gibt und die Daten nach dem Absenden nicht weiter kontrolliert werden. Somit können die Daten auch im Netzwerk verloren gehen und es wird nicht bemerkt.

Der einfache Header des UDP Protokolls besteht nur aus vier Attributen. Diese jeweils 16 Bit großen Felder enthalten den Quellport, den Zielport, die Checksumme zur Überprüfung des Inhalts und die Länge des gesamten Pakets. Insgesamt ist der Header somit 8 Byte groß. (vgl. [1][2][3])

3.2 TCP

TCP ist ein verbindungsorientiertes Protokoll, dass im OSI sieben Schichten Modell auf der vierten Ebene (Transport) einzuordnen ist. Im sogenannten TCP/Internet Protokoll (IP) Protokollstapel ist es in der dritten von vier Schichten zu finden. Bei einer Übertragung von Daten über TCP übergibt die genutzte Anwendung den Datenstrom an das Protokoll und empfängt ihn auch wieder von dort. Für die Übertragung ist dementsprechend TCP zuständig. Die Hauptaufgaben des Protkolls sind daher die Aufteilung und die Zusammensetzung der Daten von entsprechend vielen Paketen (Segmentierung), das Management der Verbindung und ein entsprechendes Fehlerhandling, welches das korrekte Empfangen von Paketen überwacht. Das Fehlerhandling nutzt eine positive Bestätigung aller Pakete. Dies bedeutet, dass nur nicht vorhandene Pakete erneut angefragt werden, ansonsten davon ausgegangen wird, dass die Daten angekommen sind. Diese Technologie sorgt dafür, dass die Daten auf jeden Fall ankommen, sofern die Verbindung nicht gestört wird. (vgl. [4][5])

Der Header eines TCP Pakets besteht aus 20 Bytes, jedoch kann dieser auch noch erweitert werden, sodass noch einige zusätzliche Bytes in den Header geschrieben werden. Zu den zwingend notwendigen Daten gehört unter anderem der Port auf dem das Paket empfangen wird und der Port über den das Paket gesendet wird. Zudem wird die Nummer im aktuellen Paketstrom benötigt. Zudem wird eine Prüfsumme und Quittierungsnummer mitgegeben, welche zur Kontrolle und Bestätigung genutzt werden. (vgl [4])

3.3 Webserver: Nginx

3.4 Arduino

Die Bezeichnung Arduino steht für eine Technologie, die sowohl aus Hardware als auch aus Software besteht. Zudem gibt es zwei Unternehmen, die in ihrem Namen den Begriff Arduino tragen. Zum einem gibt es die Arduino LLC, die die Gruppe der Gründer der Plattform bezeichnet. Zudem gibt es die Arduino S.r.l., was die Firma bezeichnet, die anfangs allein

die Arduinoboards produzierte und dann auch verkaufte. Die Arduinoplattform wurde ursprünglich entwickelt, um Beginnern den Einstieg in die Mikrokontrollerprogrammierung zu vereinfachen. Grundsätzlich besteht die sogenannte Arduinoplattform alber sowohl aus der Hardware als auch der Software. Die Hardware umfasst mittlerweile verschiedene Mikrokontroller beziehungsweise Boards, welche für verschiedene Anwendungszwecke genutzt werden können. Die entsprechende Nutzung wird mithilfe der richtigen Programmierung erreicht. Dafür kann der Softwareteil der Lösung genutzt werden, welches aus einer Integrated Development Environment (IDE) besteht und das Schreiben von Programmen vereinfacht. Zudem wird über diese IDE auch die Kommunikation mit dem Mikrokontroller realisiert. Diese Entwicklungsumgebung eignet sich für diverse Mikrokontroller, sofern entsprechende Treiber verfügbar sind, können auch Boards genutzt werden, die nicht direkt mit Arduino zusammenhängen. Zudem kann in verschiedenen Programmiersprachen entwickelt werden, beispielsweise C oder C++. (vgl. [6, 7, 8, 9, 10, 11]) Die komplette Plattform ist open source, auch wenn die Hardware natürlich bezahlt werden muss.

3.5 Python

3.6 WLAN-Chipsatz

3.7 Frontendtechnologien

3.8 REST

[12]

4 Beschreibung der Hardware

In den folgenden Kapiteln wird die ausgewählte Hardware des Projektes vorgestellt. Dabei wird zuerst auf die technischen Merkmale eingegangen und dann auf den genauen Verwendungszweck im Projekt.

4.1 Raspberry PI

4.1.1 Vorstellung des Raspberry PI

Das Raspberry PI ist ein Einplatinencomputer, den es in verschiedenen Ausführungen gibt. Je nach Ausführung variieren die Ausstattungsmerkmale. Zu den Grundsätzlichen Ausstattungsmerkmalen gehören eine CPU, unterschiedlich viel RAM und eine iGPU Einheit. Er wurde von der Raspberry PI Foundation entwickelt und hatte das ursprüngliche Ziel einen günstigen Computer für den Schulunterricht bereitzustellen. Daher ist es auch möglich eine vollständige Linux Distribution als Betriebssystem zu nutzen und es wurde sogar eine speziell angepasste Distribution veröffentlicht, die Raspbian bezeichnet wird. Aufgrund der verschiedenen Möglichkeiten wird er aber mittlerweile auch in vielen anderen Anwendungsgebieten genutzt. Insbesondere die neueren Modelle, die mit mehreren USB Ports, General Purpose Input/Output (GPIO) Pins, einem Ethernet Port und weiteren Anschlüssen ausgestattet sind, werden auch in verschiedensten Projekten privater Personen genutzt. Ein weiteres Merkmal ist die Größe des Raspberry PI's, die mit maximal 93mmx63.5mmx20mm sehr klein ausfällt. Zusätzlich gibt es diverses, bereits für den Raspberry PI ausgelegtes, Zubehör, welches weitere Erweiterungsmöglichkeiten bietet. So gibt es Kameras, Gehäuse, kleine Displays und WLAN Sticks, die den Funktionsumfang erweitern. So wurden bereits diverse Projekte vorgestellt, die zeigen, dass die Einsatzmöglichkeiten des Raspberry Pis wesentlich größer sind. (vgl. [13] [14])

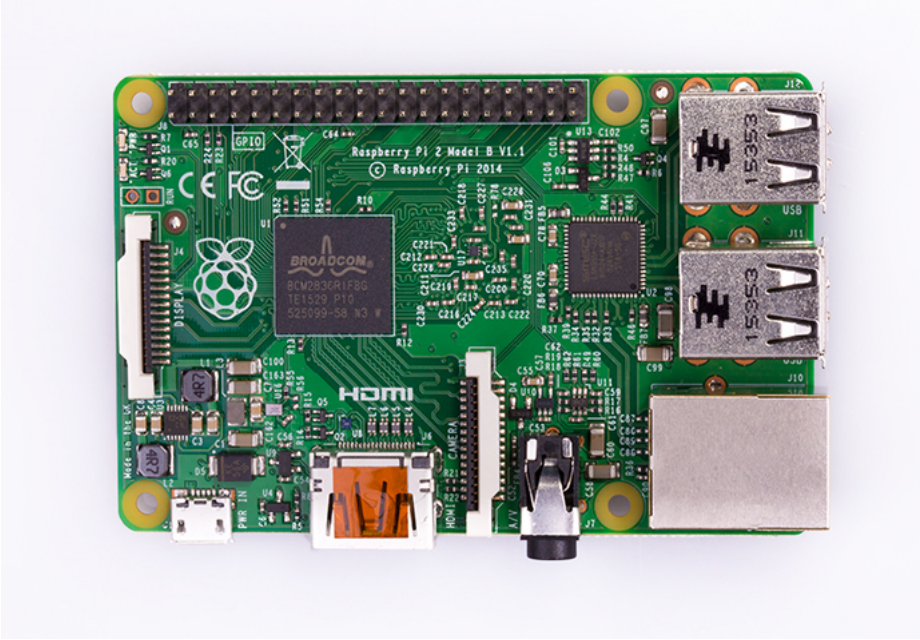

Abbildung 1: RaspberryPi Modell 2 B+,
Quelle: https://www.raspberrypi.org/wp-content/uploads/2016/02/Raspberry-Pi-2-web.jpg

4.1.2 Verwendung im Projekt

Der Raspberry PI wird für das Projekt genutzt, um einen zentralen Server bereitzustellen. Aufgrund der technischen Merkmale kann zeitgleich ein Webserver und ein Datenbankserver betrieben werden. Zudem kann über die USB Anschlüsse ein WLAN Stick angeschlossen werden, sodass die Buttons über das WLAN Netzwerk des Raspberry PIs zum zentralen Server kommunizieren können. Dazu muss mithilfe eines Skriptes der WLAN Stick von einem Empfänger zu einem Sender bzw. Access Point umfunktioniert werden. Um die dann eingehenden Nachrichten der Buttons empfangen zu können, muss zudem noch ein entsprechendes Skript im Hintergrund laufen, dass die Daten empfängt. Aufgrund von mehreren parallel laufenden Prozessen (Webserver, Datenbankserver, WLAN Skript und Skripte zum Empfangen der Daten) wird einiges an Rechenleistung benötigt, die der Raspberry Pi allerdings aufbringen kann.

Der Webserver auf dem Raspberry Pi wird dabei sowohl für das Frontend als auch für das Backend benötigt werden. Das Frontend soll dem Nutzer die Möglichkeit geben, die Liste von Waren zu verwalten und die Buttons zu konfigurieren bzw. Hilfestellung zur Einrichtung zu geben. Das Backend des Servers wird unter anderem aus einer REST Application Programming Interface (API) bestehen, die sowohl die Anfragen des Webservers verarbeitet als auch die Anfragen an die Datenbank im allgemeinen. Der Datenbankserver im Hintergrund wird die benötigte Datenbank entsprechend verwalten.

4.2 “Pretzelboard”

4.2.1 Vorstellung des “Pretzelboard”

Das “Pretzelboard” ist ein sogenanntes Elektronikmodul, dass unter verschiedenen Namen bekannt ist. Neben “Pretzelboard” ist der Name “NanoESP-Board” ebenfalls bekannt. Die Besonderheit des Boards ist ein bereits angeschlossenes WLAN Modul, welches sowohl als Sender als auch Empfänger arbeiten kann. Da es zudem den gleichen Microcontroller wie das bekanntere Microcontrollerboard “Arduino Uno” nutzt, kann die Entwicklungsumgebung von Arduino zur Entwicklung von Software genutzt werden. Der Vorteil der bereits erfolgten Kombination und Verdrahtung von WLAN Modul und Microcontroller liegt darin, dass der Nutzer dies nicht mehr machen muss und somit wesentlich weniger Kenntnisse vorausgesetzt werden müssen. (vgl. [15][16][17])

4.2.2 Verwendung im Projekt

Das “Pretzelboard” wird im Projekt als Button genutzt. Sowohl die kleine Größe als auch die bereits vorhandene WLAN Funktionalität sorgen dafür, dass sich das Board dafür besonders gut eignet. Aufgrund der Tatsache, dass es sich bezüglich der Programmierung nicht von einem Arduino Board unterscheidet, besteht die Möglichkeit, dass auf das Wissen und den Support einer bereits größeren Community zurückgegriffen werden kann. Da sich das Board zudem auf ein Elektroniksteckboard setzen lassen kann, ist auch die Verbindung mit einem Button und Statusleuchten möglich. Nach der erfolgreichen Montage kann dann das entsprechende Programm aufgespielt werden und bei vorhandener Stromversorgung kann die entsprechende Nachricht über das WLAN Netzwerk an den Empfänger gesendet werden. Das Pretzelboard kommuniziert diese Nachricht mithilfe des zuvor bereits erwähnten Prokotolls UDP.

4.3 ESP8266 Lua

4.3.1 Vorstellung des ESP8266 Lua

Der ESP8266 Lua ist ein Entwicklungsboard, welches bereits ein integriertes WLAN Modul besitzt und mit einem TCP/IP Stack ausgestattet ist. Einer der vorgesehenen Einsatzzwecke ist unter anderem das Internet of Things. Das Board kann zudem mit Steckbrett arbeiten und bietet daher ähnliche Möglichkeiten, wie das bereits erwähnte "Pretzelboard". Zudem ist es möglich, dass die Software für das ESP8266 Lua Entwicklungsboard ebenfalls über die Arduino IDE entwickelt werden kann. Allerdings sind technische Unterschiede zum Pretzelboard vorhanden, was dazu führt, dass andere Treiber genutzt werden müssen. Diese können nachinstalliert werden und stehen dann zur Nutzung des Boards bereit. (vgl. [18][19])

4.3.2 Verwendung im Projekt

Das Board wird im Rahmen des Projektes als weiterer Button genutzt. Es bietet sich für diese Funktion an, da es ebenfalls recht klein ist und alle benötigten Funktionen bereits vorhanden sind. Neben den vorhandenen Funktionen ist auch die Möglichkeit vorhanden, dass eine bereits durch das Bretzelboard bekannte Entwicklungsumgebung genutzt werden kann. Das bietet die Möglichkeit, dass statt des UDP Protokolls das bereits erwähnte TCP Protokoll ausprobiert werden kann. Der Aufbau des Boards soll ebenfalls durch ein Elektroniksteckboard umgesetzt werden. Dazu wird das Board auf eben dieses Steckboard gesetzt und mit einem Button, Statusleuchten und entsprechenden Kabeln verbunden. Nach der Betätigung des Buttons wird das Board geweckt und eine Funktion schickt ein entsprechendes Datenpaket an einen Empfänger.

4.4 Amazon Dash Button

4.4.1 Vorstellung des Amazon Dash Buttons

4.4.2 Untersuchung des Amazon Dash Buttons

4.4.3 Verwendung des Amazon Dash Buttons im Projekt

5 Umsetzung des Projektes

5.1 Architektur und Zusammenarbeit der Komponenten

Mithilfe der Technologien und Softwarelösungen, die in Kapitel 3 erläutert worden sind, und der Hardware, die in Kapitel 4 vorgestellt wurde, konnten die verschiedenen Komponenten, die bereits in der Aufgabenstellung in Kapitel 2 genannt worden sind, umgesetzt werden. Im folgenden soll ein Überblick über die Zusammenarbeit der einzelnen Komponenten gegeben werden, bevor in den folgenden Kapiteln auf die genaue Entwicklung der einzelnen Komponenten eingegangen wird.

Für den Nutzer gibt es ein Webfrontend, welches mithilfe eines Nginx Webservers realisiert wird. Dieses Webfrontend greift auf ein REST API-Backend zurück, welches in PHP: Hypertext Preprocessor (PHP)v geschrieben wurde und ebenfalls über den Nginx realisiert wird. Mithilfe dieses Backends werden die verschiedenen API Routen realisiert, die dann den Zugriff auf die Datenbank organisieren. Das Backend besteht allerdings ebenfalls aus einigen Skripten, die in Python geschrieben sind. Diese Skripte organisieren die Kommunikationsendpunkte für die Buttons indem sie auf die entsprechende Kommunikationsports für die Datenpakete lauschen und die Daten dann verarbeiten.

Neben der Kommunikation wird ebenfalls ein Teil der REST API in Python realisiert. Dieser Teil ermöglicht sowohl den Status der Ports abzufragen als auch die Skripte neu zu starten, sollte ein Skript nicht gestartet sein oder ein Fehler aufgetreten sein. Diese Daten werden ebenfalls für das Webfrontend verwendet. Die Buttons als weitere Komponente werden durch verschiedene Hardwareelemente realisiert. Mithilfe entsprechender Programme, die auf die Controller geladen werden, stellen sie eine Verbindung zu den definierten Kommunikationspunkten her. Über dieses Kommunikationsprotokoll werden dann, sobald der Button betätigt wurde, die entsprechenden Daten gesendet und dann im Backend verarbeitet.

5.2 Entwicklung der Buttons

Im folgenden soll auf die Entwicklung der Button Komponente im Projekt eingegangen werden. Dabei wird insbesondere auf die Programmierung und Umsetzung eingegangen werden. Bei der Entwicklung der verschiedenen Buttons wurde insbesondere darauf geachtet, dass verschiedene Technologien getestet werden, um möglichst viele technische Möglichkeiten zu untersuchen. Das ist damit zu begründen, dass das Untersuchen von möglichst vielen Möglichkeiten zum Projektziel gehörte.

5.2.1 Entwicklung mit dem "Pretzelboards"

5.2.2 Entwicklung mit dem ESP8266 Lua

5.2.3 Einbindung des Amazon Dash Buttons

5.3 Entwicklung des Frontends

5.3.1 Aufbau und Entwicklung des Frontends

5.4 Entwicklung und Einrichtung des Backends

In den folgenden Kapiteln wird auf die Einrichtung des Backends eingegangen. Dies umfasst sowohl die Einrichtung des Raspberry Pis, den Aufbau und die Entwicklung der REST API und die Entwicklung der Python Skripte. Die Einrichtung des Raspberry PIs wird an dieser Stelle geführt, da er das zentrale Hardwareelemente im Backend ist, da er genutzt wird, um die entsprechende Softwaredienste zur Verfügung zu stellen. Er stellt den zentralen Kommunikationspunkt im Projekt dar, da die Buttons nur ihn kennen, sich aber nicht gegenseitig.

5.4.1 Einrichtung des Raspberrys

Die Einrichtung des Raspberry PIs besteht aus mehreren Schritten an dessen Ende die Verwendung des Raspberrys als zentraler Server steht. Die verschiedenen Schritte werden im folgenden erklärt:

Einrichtung des Nginx

Einrichtung des SQL Datenbankservers

Einrichtung des WLAN Access Points

Einrichtung von Python:

Im Rahmen der Einrichtung des Raspberry Pi's ist auch die Einrichtung von Python durchzuführen. Für eine erfolgreiche Einrichtung müssen die entsprechenden Softwarepakete installiert werden. Neben der grundlegenden Instalaltion von Python galt es auch die entsprechenden Skripte zu entwickeln. Dies wird allerdings genauer im Kapitel 5.4.3 betrachtet.

5.4.2 Aufbau und Entwicklung der REST API

5.4.3 Aufbau und Entwicklung der Python Skripte

Mithilfe der Skriptsprache Python werden die verschiedenen Kommunikationsendpunkte in Form von Sockets realisiert. Um eine bessere Übersichtlichkeit zu gewährleisten und eine zentrale Verwaltung zu haben, gibt es ein Verwaltungsskript. Dieses Skript startet die anderen Skripte, die sich um jeweils einen Kommunikationsprotokoll kümmern. So gibt es ein Skript für die Kommunikation über UDP, eins für TCP und eins für Address Resolution Protocol (ARP), welches für die Amazon Dash Buttons genutzt wird. Aus diesen Gründen gibt es insgesamt vier Python Skripte, die einen wesentlichen Teil des Backends ausmachen.

Entwicklung des Verwaltungsskripts

Das Verwaltungsskript dienst, wie bereits erwähnt, als zentrales Skript, welches als einziges Skript auch gestartet werden muss. Über dieses Skript werden dann alle anderen notwendigen Skripte gestartet, die dann dafür sorgen, dass die Kommunikation ermöglicht wird. Neben dieser Funktionalität enthält das Verwaltungsskript auch die Teile der REST API, die über Python realisiert wird. Dies sind einige Methoden, die bestimmte Adressen dafinieren. Mithilfe dieser Adressen kann abgefragt werden, ob die entsprechenden Kommunikationsskripte noch laufen. Die Antwort wird als JavaScript Object Notation (JSON) String zurückgeliefert und kann dann entsprechend weiterverarbeitet werden.

Entwicklung des UDP Skripts

Da die Übertragung der Datenpakete über das UDP Protokoll ermöglicht werden soll, muss ein entsprechender Empfänger auf dem Raspberry PI vorhanden sein. Dieser Empfänger wird mithilfe eines Skriptes in Python realisiert.

Dieses Skript nutzt die Library "Socket" (vgl. [20]), welches es ermöglicht ein Socket zu erstellen. Dieses Socket wird an eine IP Adresse und einen Port gebunden und wird anschließend für alle eingehenden UDP Pakete genutzt. In einer Endlosschleife, welche manuell unterbrochen werden kann, wird auf eingehende Pakete gewartet. Die Endlosschleife wird benötigt, da ein Button zu jedem Zeitpunkt betätigt werden kann und somit dauerhaft auf ankommende Pakete geachtet werden muss.

Bei Eingang eines entsprechenden Pakets wird ein entsprechendes Request an die REST API geschickt. Da als Übertragungsart allerdings das UDP Protrokoll genutzt wird, kann dem Button keine Rückmeldung gegeben werden, ob der Eintrag in die Datenbank über die REST API erfolgreich war. Zudem kann auch kein allgemeines Feedback zurückgegeben werden. Auch bei anderen Fehlern kann dem Button keine Rückmeldung gegeben werden.

Nach dem erfolgreichen Absenden der Anfrage an die REST API befindet sich das Skript für den UDP Empfänger weiterhin in der Endlosschleife und wartet auf das nächste Datenpaket.

Entwicklung des TCP Skripts

Für alle Datenpakete, die über das Protokoll TCP empfangen werden, muss ebenfalls ein Skript geschrieben werden, welches diese Datenpakete verarbeitet. Dabei wird genauso vorgegangen, wie beim Skript für UDP. Der einzige Unterschied ist nach dem Absenden der Anfrage an die REST API. Das Skript für TCP Datenpakete wartet nach dem Absenden der Anfrage auf die Bestätigung und schickt eine entsprechende Antwort zurück an den Button. Dieser verarbeitet ebenfalls die Antwort und kann mithilfe einer Lampe dem Nutzer ein visuelles Feedback geben.

Entwicklung des ARP Skripts

6 Ergebnis des Projektes

7 Fazit und Ausblick

8 Anhang

8.1 Skripte

UDP Skript für den Raspberry PI:

```
import subprocess
import socket
import MySQLdb
import requests
import json

db = MySQLdb.connect("localhost", "tobias", "tobias", "temps")
curs = db.cursor()
subprocess.call(["echo", "DB connected"]);


sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(('192.168.1.1',8888))
#s.listen(0)
#s.accept()
while True:
        data, addr = sock.recvfrom(1024)
        print "received message:", data
        sql = "INSERT INTO test(ID, MESSAGE) VALUES ('1','%s')" % \
              (data)

        url = "http://localhost/api/products"
        data = {'name': 'Test', 'price': 0.1}
        headers = {'Content-type': 'application/json', 'Accept': 'text/pla
        r = requests.post(url, data=json.dumps(data), headers=headers)
        print r.text

        try:
                curs.execute(sql)
                db.commit()
        except:
                print 'Fehler'
```

9 Abkürzungsverzeichnis

API Application Programming Interface

ARP Address Resolution Protocol

GPIO General Purpose Input/Output

IoT Internet of Things

IDE Integrated Development Environment

IP Internet Protokoll

JSON JavaScript Object Notation

OSI Open Systems Interconnection

PHP PHP: Hypertext Preprocessor

REST Representational State Transfer

TCP Transmission Control Protocol

UDP User Datagram Protocol

Literatur

[1] Elektronik-Kompendium. Tcp - transmission control protocol. http://www.elektronik-kompendium.de/sites/net/0812271.htm.

[2] Elektronik-Kompendium. Udp - user datagram protocol. http://www.elektronik-kompendium.de/sites/net/0812281.htm.

[3] ITwissen.info. Udp :: user datagram protocol :: Udp-protokoll :: Itwissen.info. http://www.itwissen.info/definition/lexikon/user-datagram-protocol-UDP-UDP-Protokoll.html, 23.02.2016.

[4] Tcp - transmission control protocol. https://www.elektronik-kompendium.de/sites/net/0812271.htm.

[5] Tcp :: transmission control protocol :: Tcp-protokoll :: Itwissen.info. http://www.itwissen.info/definition/lexikon/transmission-control-protocol-TCP-TCP-Protokoll.html, 22.11.2016.

[6] Arduino - environment. https://www.arduino.cc/en/Guide/Environment.

[7] Arduino - getting started. https://www.arduino.cc/en/Guide/HomePage.

[8] Arduino - getting started. https://www.arduino.cc/en/Guide/HomePage.

[9] Arduino - introduction. https://www.arduino.cc/en/Guide/Introduction.

[10] Getting started. http://www.arduino.org/learning/getting-started.

[11] heise online. Arduino - mikrocontroller für einsteiger. https://www.heise.de/thema/Arduino.

[12] Stefan Tilkov, Martin Eigenbrodt, Silvia Schreier, and Oliver Wolf. *REST und HTTP: Entwicklung und Integration nach dem Architekturstil des Web*. dpunkt, s.l., 3. aufl. edition, 2015.

[13] Rpi hardware - elinux.org. http://www.elinux.org/RPi_Hardware, 28.12.2016.

[14] Raspberry Pi Foundation. Raspberry pi foundation - about us. https://www.raspberrypi.org/about/, 28.01.2017.

[15] Wifi board: Nanoesp bzw. pretzel-board. http://www.mikrocontroller-elektronik.de/wifi-board-nanoesp-bzw-pretzel-board/.

[16] Technische daten nanoesp & pretzel board. http://iot.fkainka.de/technische-daten.

[17] Franzis Verlag GmbH and München. Pretzel-board. http://www.franzis.de/elektronik/arduino-platinen/pretzel-board, 27.11.2015.

[18] Frank Carius. Msxfaq.de:nodemcu. http://www.msxfaq.de/sonst/bastelbude/nodemcu.htm, 15.01.2017.

[19] Elegiant nodemcu lua esp8266 esp-12e wifi development: Amazon.de: Computer & zubehör. https://www.amazon.de/ELEGIANT-NodeMcu-ESP8266-ESP-12E-Development/dp/B018E741G4/ref=sr_1_1?ie=UTF8&qid=1487860153&sr=8-1&keywords=esp8266+lua.

[20] 18.1. socket — low-level networking interface — python 3.6.0 documentation. https://docs.python.org/3/library/socket.html, 20.02.2017.

Abbildungsverzeichnis