

Ausarbeitungen und Poster zum Seminar

**Aus alt mach neu:
Upcycling mit Raspberry Pi und Co.**

Sommersemester 2019

15. Juli 2019



Fakultät II – Informatik, Wirtschafts- und
Rechtswissenschaften
Department für Informatik
Abt. Systemsoftware und verteilte Systeme



www.uni-oldenburg.de/svs



Inhaltsverzeichnis

1 Upcycling eines Radioweckers (Marius Lewrick)	1
2 Poster (Marius Lewrick)	7
3 Proseminar - Aus alt mach neu: Upcycling mit Raspberry Pi und Co. (Janek de Buhr)	8
4 Poster (Janek de Buhr)	11
5 Aus alt mach neu: Upcycling einer Senseo-Kaffeemaschine mit dem Raspberry Pi (Arne Schlömer)	12
6 Poster (Arne Schlömer)	17
7 Upcycling eines Briefkastens (Anatolij Fandrich)	18
8 Poster (Anatolij Fandrich)	22
9 Raspberry Pi VoIP Softphone (Sören Rempel)	23
10 Poster (Sören Rempel)	26
11 Spiegel mit Wetteranzeige durch LED-Hintergrundbeleuchtung (Sarah Katharina Thien)	27
12 Poster (Sarah Katharina Thien)	32
13 Aus alt mach neu - Aufwerten einer Thermoskanne mithilfe eines Wemos D1 mini (Jana Gerdes)	33
14 Poster (Jana Gerdes)	38
15 DIY-Longboard mit Gewichtverlagerung (Dimitri Manschos)	39
16 Poster (Dimitri Manschos)	42
17 Kaminuhr 2.0 – Upcycling einer Kaminuhr (Florian Schmalriede)	43
18 Poster (Florian Schmalriede)	49

19 Erweitertes digitales Barometer zur Erfassung von Schimmel (Niklas Holtz)	50
20 Poster (Niklas Holtz)	55
21 Upcycling mit Raspberry Pi ud Co.: Internetradio (Mykhailo levreinov)	56
22 Poster (Mykhailo levreinov)	59
23 Upcycling eines ferngesteuerten Autos (Paula Bürmann)	60
24 Poster (Paula Bürmann)	64

Upcycling eines Radioweckers

Marius Lewrick
Department für Informatik
Carl von Ossietzky Universität Oldenburg

Abstract

In dieser Arbeit geht es darum, einen alten Radiowecker in seiner Funktionalität zu erweitern. Der Wecker soll den Benutzer automatisch und rechtzeitig auf Grundlage eines gegebenen Terminkalenders wecken. Hierbei soll der Benutzer die Möglichkeit haben einige Einstellungen vornehmen zu können, um den smarten Wecker an das eigene Verhalten anzupassen. So kann zum Beispiel der Zeitraum eingestellt werden, in dem der Wecker Termine berücksichtigt.

1 Einleitung

Im Zuge des Proseminars “Aus alt mach neu - Upcycling mit dem Raspberry Pi & Co.“ sollen Geräte, welche insbesondere technisch veraltet sind, mit neuen, zeitgemäßen Funktionen ausgestattet werden. Die Funktionalitäten sollen hierbei mit Hilfe eines Einplatinen-Computers wie dem Raspberry Pi oder Microcontrollern wie dem Arduino Uno realisiert werden. Beispielsweise ist es dadurch möglich, bei einem alten Radio ein Upcycling durchzuführen, sodass es zusätzlich zu normalen Radiosendern auch Web-Radiosender empfangen und abspielen kann.

1.1 Entwicklung des Weckers

Die Geschichte des Weckers beginnt zwar schon in der Antike, aber wir betrachten in dieser Arbeit die Entwicklung seit dem mechanischen Wecker. Diesen konnte man über mehrere Räder (Kronen) an der Rückseite einstellen. Mit einem dieser Räder wurden die Uhrzeiger bewegt und somit die Zeit eingestellt, ein anderes war dazu da um die Weckzeit einzustellen und das dritte wurde benötigt um das Uhrwerk aufzuziehen. Der Wecker konnte durch das Aufziehen mit einem Federwerk betrieben werden. Zum Wecken wurden dann Glocken auf dem Wecker verwendet. Die Anzeige der Uhrzeit wird hierbei über ein analoges Ziffernblatt realisiert.

Als nächstes war es möglich, Wecker mit Elektrizität über die Steckdose oder mit Hilfe einer Batterie zu betrei-

ben. Solche Wecker können die Uhrzeit digital über ein Display ausgeben. Die Steuerung des Alarms kann durch einen “Digital-Uhr-IC “ (z.B. einen SC8560S) über Taster eingestellt werden.

Ab diesem Zeitpunkt haben Wecker nur noch weitere zusätzliche Eigenschaften erhalten, welche allerdings keine nennenswerten Auswirkungen auf die Grundfunktion haben. So gibt es zum Beispiel Radiowecker, die neben der Alarmfunktion und der Anzeige der Uhrzeit auch als einfaches Radio genutzt werden können. Des Weiteren ist auch die Schlummerfunktion eine solche Funktion. Wird Schlummer-Taste während des Alarms genutzt, so stoppt dieser und wird nach einer vorgegebenen Zeit fortgesetzt bzw. wiederholt [1].

1.2 Upcycling-Möglichkeiten eines einfachen Weckers

Um nun die Upcycling-Möglichkeiten zu betrachten, gehen wir zunächst von einem einfachen Wecker aus, welcher außer einer Alarmfunktion keine weiteren Funktionen besitzt. Eine Möglichkeit, einen solchen Wecker zu verbessern, wäre, die Radiofunktion die schon in Abschnitt 1.1 genannt wurde. Hierbei ist es zeitgemäß, wenn dieses Radio auch Internet-Radiosender abspielen kann.

Eine weitere Option ist es, den Wecker mit einem Aufwach-Licht auszustatten. Dabei handelt es sich um ein Licht, welches sukzessive heller wird, um im dunklen Schlafzimmer einen Sonnenaufgang zu simulieren. Der Benutzer wird damit nicht mehr durch ein störendes Geräusch abrupt geweckt, sondern langsam durch die steigende Lichtintensität im Raum.

In der heutigen Zeit wird der klassische Wecker allerdings häufig durch Geräte wie den Amazon Echo Spot [2] oder ein Smartphone ersetzt. Diese Geräte haben einen weitaus größeren Funktionsumfang, da sie nicht nur für das Aufwecken des Benutzer konzipiert wurden. Es ist auch möglich einen alten Wecker teilweise mit Funktionen eines Smartphones oder ähnlichem auszustatten. So kann eine Erweiterung um Smart-Home-Funktionen stattfinden. Beispielsweise kann beim Alarm auch gleichzeitig das Licht

an gehen und/oder die Kaffeemaschine angeschaltet werden. Auch ein Display zur Anzeige von Informationen wäre möglich. Die Bedienung kann über einen Touchscreen oder eine Sprachsteuerung realisiert werden, was eine weitere Upcycling-Möglichkeit wäre.

1.3 Idee: Der Smarte Wecker

Auch in dieser Arbeit soll ein Wecker einem Upcycling unterzogen werden. Als Ausgangsgerät dient dabei ein alter Radiowecker. Ziel des Upcyclings ist es, diesen Wecker "smart" zu machen. D.h., das Gerät soll den Benutzer automatisch, auf Grundlage eines Kalenders, rechtzeitig wecken. Vorteil hierbei ist, dass der Benutzer sich keine Gedanken mehr darüber machen muss, ob der Wecker gestellt worden ist oder nicht. Sobald der Terminkalender richtig gepflegt ist, passiert alles automatisch.

Es soll zusätzlich möglich sein, einige Schwellwerte am Wecker zu konfigurieren. Ein Beispiel wäre hier, wie viele Minuten vor dem Termin der Alarm ausgelöst werden soll. Dazu soll ein Intervall einstellbar sein, in dem Termine vom Wecker berücksichtigt werden (z.B. nur Termine zwischen 6 und 10 Uhr).

2 Hauptteil

Im Folgenden wird die Idee des "Smarten Weckers" konkretisiert, indem zunächst das Ausgangsgerät analysiert wird. Anschließend wird ein Konzept für das Upcycling erstellt und die Vorgehensweise beim Aufbau detailliert beschrieben.

2.1 Aufbau des Ausgangsgeräts

Als Ausgangsgerät dient ein "Radio Alarm Clock KH20", bei welchem es sich um einen Radiowecker mit einfachem Funktionsumfang handelt. Es ist lediglich möglich, die Uhrzeit und eine Weckzeit einzustellen. Zudem verfügt der Wecker über eine Schlummerfunktion, und kann als Radio genutzt werden.



Abbildung 1. Radio Alarm Clock KH20

Im Inneren des Radioweckers befindet sich ein Digitaluhr-IC vom Typ SC8560S [3]. Dieser IC dient zur Kontrolle des gesamten Gerätes und bildet die Verbindung zwischen Benutzereingabe, Display und Radio.

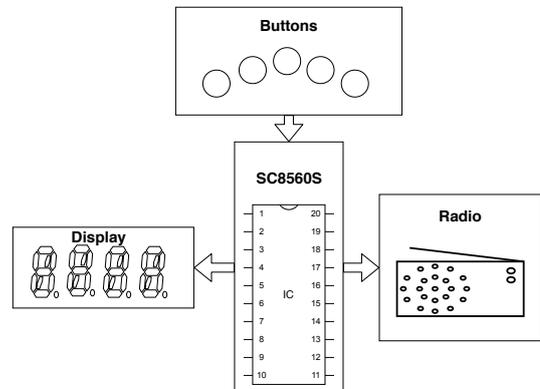


Abbildung 2. Aufbau des Ausgangsgeräts

In Abbildung 1 ist zu erkennen, dass der Wecker durch einen IC, welcher über die Tasten konfiguriert werden kann bedient wird. Der IC sorgt dann dafür, dass die Zeit auf dem Display angezeigt wird und das bei Alarm oder beim Einschalten der Radio-Funktion eine entsprechende Ausgabe am Lautsprecher stattfindet.

2.2 Konzeptioneller Aufbau des smarten Weckers

Um das Upcycling im Wecker umzusetzen, wird ein Raspberry Pi 3 genutzt. Dabei gibt es zwei Möglichkeiten, das Projekt umzusetzen:

Option 1: Raspberry Pi in bestehenden Schaltkreis einbauen

Das Raspberry Pi steuert den Wecker mithilfe der GPIO-Pins direkt am bestehenden Schaltkreis. Hierfür müsste der Raspberry Pi anstelle des ICs agieren. Der Vorteil wäre dann, dass man die bestehenden Komponenten des Weckers mitbenutzen könnte, sodass kein Lautsprecher oder Taster mehr benötigt wird. Allerdings nutzt der Wecker eine andere Spannung, sodass Spannungswandler nötig wären.

Option 2: Pi stellt Funktion autark bereit

Hierbei funktioniert das Raspberry Pi unabhängig vom Wecker. Bei dieser Variante wird zusätzlich ein Lautsprecher sowie ein Taster zur Steuerung benötigt. Dafür kann das System in jedem anderen Wecker ebenfalls Problemlos genutzt werden, indem es in das Gehäuse eingebaut wird.

Zudem bietet diese Lösung eine höhere Flexibilität, da keine GPIO-Pins für feste Pins des ICs genutzt werden, welche bei anderen Weckermodellen mit ggf. unterschiedlichen ICs nicht vorhanden sind.

Aufbau mit Option 2

Aufgrund der höheren Flexibilität und der dadurch möglichen Wiederverwendbarkeit des Systems wird in diesem Projekt die zweite Option verwendet. Das heißt, die zusätzliche Funktionalität wird unabhängig vom Ausgangsgerät realisiert.

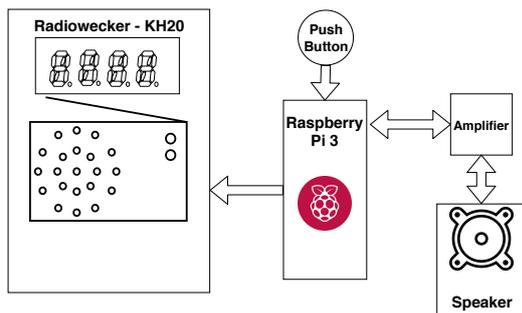


Abbildung 3. Aufbau mit Raspberry Pi

In Abbildung 2 wird deutlich, dass ein komplettes System in den vorhandenen Wecker eingebaut wird. Um die Idee umzusetzen, wird neben dem Raspberry Pi ein Button zum Ausschalten des Alarms benötigt. Außerdem muss für einen ausreichend lauten Lautsprecher ein Verstärker an das Raspberry Pi angeschlossen werden. Der interne 3,5 mm-Klinkenanschluss des Raspberry Pis ist nur für Kopfhörer ausgelegt und bietet daher nicht genügend Leistung für größere Lautsprecher.

Softwarekonzept

Damit der Raspberry Pi die gewünschte Funktionalität bereitstellen kann, muss ein entsprechendes Programm entwickelt werden. Dazu benötigt der Benutzer eine Möglichkeit, den smarten Wecker zu konfigurieren. Hier bietet es sich an, eine Web-Applikation zu programmieren, sodass der Benutzer alles bequem über eine Web-Oberfläche steuern kann.

In diesem Projekt soll die Web-Applikation mit Hilfe des Python-Frameworks "Flask" [4] realisiert werden. Dabei bietet Python[5], über diverse Module, alle für die Applikation notwendigen Funktionen. Um zu ermöglichen, dass alle Einstellungen auch nach Neustart des Rasperrys erhalten bleiben, wird für die Applikation eine Config-Datei zur Verfügung gestellt.

Da der Weckalarm immer auf Basis von Terminen aus einem Kalender ausgelöst werden soll, wird die "Google Calendar API"[6] verwendet. Damit hat der Nutzer die Möglichkeit, sich mit seinem Google-Account anzumelden, sodass der Wecker die Termine des Google-Kalenders nutzen kann.

Zusätzlich muss auf dem Raspberry Pi ein Web-Server für die Web-Applikation und ein Treiber für den Verstärker installiert werden, damit der Lautsprecher funktioniert.

Das gesamte Konzept wird in Abbildung 3 dargestellt. Es ist erkennbar, dass die Flask-Applikation Termininformationen aus dem Google-Calendar erhält, und die notwendigen Einstellungen vom Benutzer durch das Frontend getätigt werden. Durch die Applikation wird der Raspberry Pi gesteuert, sodass dieser im Falle eines Alarms etwas am Lautsprecher ausgibt. Der Push-Button kann genutzt werden, um den Alarm wieder auszuschalten.

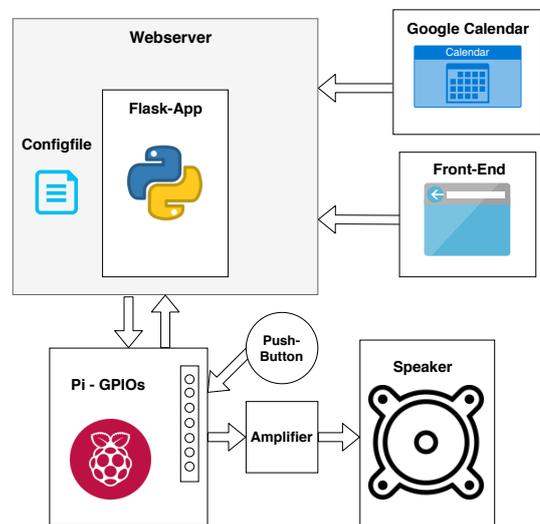


Abbildung 4. Softwarekomponenten

2.3 Finale Realisierung

In diese Kapitel werden Hardwareaufbau sowie die Softwareimplementierung genau betrachtet.

Hardwareaufbau

Für den Aufbau wird folgende Hardware benötigt:

- Raspberry Pi 3
- Netzteil (5 Volt)
- SD-Karte
- Adafruit MAX98357A Verstärker [7]

- Lautsprecher (4Ω, 3W)
- Taster
- Kabel zum Verbinden

Der gesamte Aufbau wird in Abbildung 5 dargestellt.

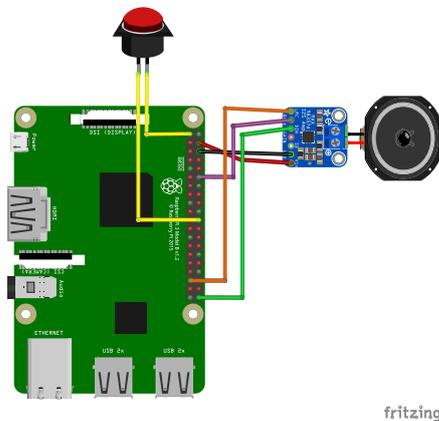


Abbildung 5. Hardwareaufbau

Die Stromversorgung des Tasters wird über den 3,3 Volt Pin des Raspberry Pis sichergestellt. Für den Verstärker wird ein 5-Volt-Pin sowie ein GND-Pin benötigt. Die übrigen Kabel werden an diverse GPIO-Pins angeschlossen. Diese Pins werden durch die Software des Verstärkers vorgegeben.

Die Stromversorgung des Raspberry Pis wird durch ein geeignetes 5-Volt-Netzteil umgesetzt. Zum sicheren Herunterfahren des Systems, kann per ssh ein Shutdown ausgelöst werden. Alternativ kann in der Web-Oberfläche der Applikation eine Schaltfläche zum Herunterfahren betätigt werden.

Softwareumsetzung

Bei der Softwareumsetzung musste zunächst die Anmeldung mit Google realisiert werden, damit die Applikation Zugriff auf die Kalenderdaten des Benutzers erhält. Dazu müssen von Google Zugangsdaten für die Applikation heruntergeladen werden. Diese Zugangsdaten werden in Form einer JSON-Datei, zusammen mit der Applikation auf dem Web-Server gespeichert. Google kann die Applikation später anhand dieser Zugangsdaten identifizieren. Zusätzlich muss der Benutzer ein durch den persönlichen Google-Account generiertes Token auf dem Server hinterlegen. Das Token erlaubt dann der Applikation, die sich mit den oben genannten Zugangsdaten (credentials.json) identifiziert, auf die Kalenderdaten des zum Token gehörenden Benutzers zuzugreifen.

```

1 if os.path.exists('token.pickle'):
2     print("Token already exists!")
3     if not creds or not creds.valid:
4         if creds and creds.expired and
5             creds.refresh_token:
6             creds.refresh(Request())
7         else:
8             flow=InstalledAppFlow.from_...
9             ... client_secrets_file(
10                'credentials.json', SCOPES)
11             creds = flow.run_local_server()
12             with open('token.pickle', 'wb')...
13                 ... as token:
14                 pickle.dump(creds, token)

```

Listing 1. getgoogletoken

In vorangehenden Codesnipped ist das Skript abgebildet, welches der Nutzer ausführen muss um an das entsprechende Token zu kommen. Ist kein Token vorhanden, wird der Benutzer zum Google-Login im Browser weitergeleitet. Dort muss sich der Benutzer mit seinem Google-Account anmelden und bestätigen, dass die Applikation, welche sich durch die Datei "credentials.json" identifiziert, Zugriff auf den Kalender erhalten darf.

Da der Benutzer für den Token-Upload selbst zuständig ist und die Web-Applikation dennoch möglichst benutzerfreundlich sein soll, wird die Routing-Funktion von Flask genutzt. Es wird eine entsprechende Anleitung angezeigt, falls kein Token auf dem Server vorhanden ist. Das Vorgehen der Applikation für das Routing, ist nachfolgend dargestellt:

```

1 @app.route('/')
2 def homepage():
3     if os.path.exists('token.pickle'):
4         value = getConfigValue(
5             'alarmconfig',
6             'buffertoevent')
7         return render_template(
8             "configure.html",
9             standard=value,
10            status="",
11            bgcolor="bg-white")
12     else:
13         return render_template(
14             "tokenupload.html",
15             alert=None,
16             text="token.pickle_hochladen!")

```

Listing 2. Flask-Routing

Wenn kein valides Google-Token auf dem Server vorhanden ist, wird der Benutzer auf die Seite "tokenupload.html" weitergeleitet. Hier bekommt der Be-

nutzer dann die Möglichkeit, das Skript aus Listing 1 zusammen mit den Applikationscredentials (credentials.json) herunterzuladen, um das Token zu generieren. Außerdem kann auf dieser Seite das Token auf den Server hochgeladen werden. Erst nachdem das passiert ist, kann der Benutzer durch eine weitere Route auf die Seite “configure.html” gelangen. Hier können dann für den Wecker relevante Einstellungen getroffen werden.

HTML-Seiten sind in Flask sogenannte Templates. Da ein einzelnes Template unter Umständen unterschiedliche Inhalte anzeigen muss, können die HTML-Seiten mithilfe einer Template-Engine mit Variablen versehen werden. Diese werden durch die Return-Statements in den Flask-Routen mit Inhalt gefüllt. Innerhalb von Flask wird die Template-Engine Jinja2 [8] genutzt. Variablen in HTML-Dateien können dadurch wie in nachfolgendem Code dargestellt werden:

```
1 <p>
2 Zeit vor Termin in Minuten:
3 <input type="text" name="time"
4 value="{{ _standard }}" />
5 </p>
```

Listing 3. Jinja2 Variable

Die Variable, die hier gesetzt wird, heißt “standard“ und gibt an, welchen Wert die Value-Option des HTML-Tags “input“ annehmen soll. In Listing 2, Zeile 16, kann man erkennen, dass dieser Wert beim Return-Statement gesetzt wird, sodass Flask das Template mit entsprechenden Inhalt laden kann. In diesem Fall wird dafür ein Wert aus einer Config-Datei geholt.

Die Einstellungen für den smarten Wecker werden in einer INI-Datei gespeichert. Zur Umsetzung dieser Funktion wird das Python-Modul Configparser [9] genutzt. Um die Config-Datei in der gesamten Applikation nutzen zu können, wurden die Methoden `getConfigValue()` zum Erhalten eines Wertes aus der INI-Datei und `setConfigValue()` zum Setzen eines Wertes in der INI-Datei implementiert. Es wird die Vorlaufzeit zum Termin und der Zeitraum in dem die Termine berücksichtigt werden gespeichert. Dazu soll der Startzeitpunkt einstellbar sein, an welchem der Wecker den nächsten Termin sucht (Initialer Job).

Jetzt muss noch die eigentliche Implementierung der Weckerfunktion vorgenommen werden. Dazu wird das Python-Modul APScheduler [10] verwendet. Mit dem APScheduler ist es möglich Cronjobs [11] im Hintergrund auszuführen. Das heißt, eine Methode kann zu einem vorher definierten Zeitpunkt ausgeführt werden. In Python sieht das dann wie folgt aus:

```
1 initjob = scheduler.add_job(initialarm ,
2 'cron' ,
3 hour=3,
```

```
4 minute=0)
```

Listing 4. Cronjob

Zunächst wird hier die Methode “initialarm“ angegeben. Danach folgt der Typ des Jobs (hier: Cron) und dann die Stunde und Minute, in welcher der Job ausgeführt werden soll. Die Initialarm-Methode sorgt dafür, dass ein weiterer Cronjob (Alarmauslöser) zur Weckzeit erstellt wird. Dies geschieht auf Grundlage des nächsten Termins am gleichen Tag, abzüglich der in der Config-Datei angegebenen Vorlaufzeit. Dieser Alarm-Job startet zur entsprechenden Zeit ein Webradio um den Benutzer zu wecken. Wird der Alarm beendet, so wird auch der Job entfernt, damit dieser am nächsten Tag kein weiteres Mal ausgelöst wird.

Zur Umsetzung des Alarms wurde eine Radio-Klasse geschrieben, welche mithilfe des VLC-Media-Players [12] arbeitet. Der VLC-Player kann über das Python-Modul `python-vlc` [13] durch die Applikation angesteuert werden. Innerhalb der Klasse gibt es Methoden zum Starten und Stoppen des VLC-Media-Players, welcher dann den vorgegebenen Web-Radiosender abspielt. Wird die `startRadio`-Methode durch den Alarm-Cronjob ausgelöst, läuft das Radio zehn Minuten lang, außer der Alarm wird durch Betätigung des am Raspberry Pi verbauten Tasters unterbrochen. Durch das Python-Modul `RPi.GPIO` [14] kann die Web-Applikation das Signal des Tasters abhören und entsprechend auf einen Tastendruck reagieren. Die Startmethode wird wie anschließend dargestellt realisiert.

```
1 def startRadio(self):
2     self.player.play()
3     counter = 0
4     while counter <= 600:
5         time.sleep(1)
6         counter += 1
7         if GPIO.input(25):
8             break
9     self.player.stop()
```

Listing 5. startRadio

Der angesprochene Taster befindet sich in diesem Fall an GPIO-PIN 25 des Raspberry Pis. Die Schleife wird durchlaufen, bis der Taster betätigt wurde oder der Counter die Zahl 600 erreicht hat.

Zuletzt wurde noch ein Button innerhalb der Web-Oberfläche realisiert, welcher ein Shutdown-Befehl auf dem Raspberry Pi auslösen kann. Dies dient zum sicheren Herunterfahren des Weckers

3 Zusammenfassung

In diesem Kapitel wird ein Ausblick auf mögliche Verbesserungen gegeben und ein Fazit aus dem bisherigen Ent-

wicklungsverlaufs des Projektes gezogen.

3.1 Ausblick: Verbesserungsmöglichkeiten

Als Verbesserungsmöglichkeiten bieten sich einige Optionen an. Hierbei werden die Punkte aus Abschnitt 1.2 nicht erneut betrachtet.

Bei der Weckzeit wird bisher ein fester Wert übergeben um zum Beispiel eine Stunde vor Terminbeginn den Alarm auszulösen. Dies könnte auch dynamisch Passieren, indem der Wecker die Zeit anhand der Fahrzeit zum Termin anpasst. Das wäre über die Google-API (Maps) möglich, allerdings müsste bei jedem Termin ein Ort angegeben werden.

Des Weiteren kann der smarte Wecker auch mit einer normalen, über die Web-Applikation steuerbaren Internetradio-Funktion ausgestattet werden. Das Radio wird dann nicht nur zum Wecken genutzt.

Eine weitere Verbesserung ist das Hinzufügen von mehr gängigen Kalenderdiensten wie iCloud-Kalender oder Exchange, damit die Anwendung dem Nutzer mehr Flexibilität bietet und dieser nicht gezwungen wird, ausschließlich Google zu nutzen.

Zusätzlich dazu könnte man eine Schlummerfunktion einbauen und eine Statistik in der Web-Applikation anzeigen. Der Nutzer könnte somit sehen, wie oft er die Funktion durchschnittlich nutzt.

3.2 Fazit

Die im Projekt umgesetzte zusätzliche Funktionalität konnte zufriedenstellend bereitgestellt werden. Der alte Radiowecker ist durch den Raspberry Pi in der Lage, den Benutzer zu seinem ersten Termin durch einen Alarm zu wecken. Lediglich die erste Einrichtung des Systems ist umständlich, da ein durch Google generiertes Token vom Benutzer auf dem Webserver des Raspberry Pis hochgeladen werden muss. Danach kann der Nutzer die Einstellungen bequem über eine Web-Oberfläche an seine Wünsche anpassen.

Dabei konnte der originale Funktionsumfang des Radioweckers erhalten bleiben, sodass dieser auch weiterhin wie ein klassischer Wecker verwendet werden kann.

Literatur

- [1] Wikipedia-Eintrag zu "Wecker":
<https://de.wikipedia.org/wiki/Wecker> (abgerufen am 26.05.2019)
- [2] Produktdaten "Amazon Echo Spot":
<https://amzn.to/2WtGDfL> (abgerufen am 09.06.2019)

- [3] Datenblatt des SC8560S ICs:
<https://datasheet4u.com/datasheet-pdf-file/1077530/SilanMicroelectronics/SC8560S/1> (abgerufen am 09.06.2019)
- [4] Flask-Dokumentation:
<http://flask.pocoo.org/docs/1.0/> (abgerufen am 09.06.2019)
- [5] Python-Dokumentation:
<https://www.python.org/> (abgerufen am 09.06.2019)
- [6] Google-Calendar-API-Dokumentation:
<https://developers.google.com/calendar/> (abgerufen am 09.06.2019)
- [7] Datenblatt des Verstärkers:
<https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/overview> (abgerufen am 09.06.2019)
- [8] Dokumentation von Jinja2:
<http://jinja.pocoo.org/docs/2.10/> (abgerufen am 10.06.2019)
- [9] Dokumentation des Python-Configparsers:
<https://docs.python.org/3/library/configparser.html> (abgerufen am 10.06.2019)
- [10] Dokumentation vom APScheduler:
<https://apscheduler.readthedocs.io/en/latest/> (abgerufen am 10.06.2019)
- [11] Wikipedia-Eintrag zu "Cron":
<https://de.wikipedia.org/wiki/Cron> (abgerufen am 10.06.2019)
- [12] VLC-Media-Player Homepage:
<https://www.vlc.de/> (abgerufen am 10.06.2019)
- [13] Dokumentation python-vlc:
<https://pypi.org/project/python-vlc/> (abgerufen am 10.06.2019)
- [14] PyPi Dokumentation: htt-
[ps://pypi.org/project/RPi.GPIO/](https://pypi.org/project/RPi.GPIO/) (abgerufen am 10.06.2019)

Upcycling eines Radioweckers

Marius Lewrick

EINEN ALTEN RADIOWECKER "SMART" MACHEN

- Funktionen des Weckers vor dem Upcycling:
 - Schlummertaste
 - Uhrzeit
 - Weck- / Alarmzeiteinstellung
 - UKW-Radio
- Funktionen die durch das Upcycling hinzugefügt werden sollen:
 - Weboberfläche für Einstellungen
 - Google-Authentifizierung durch Token
 - automatischer Alarm auf Grundlage des ersten Googlekalender-Termins am Tag
 - Webradiosender als Wecksound



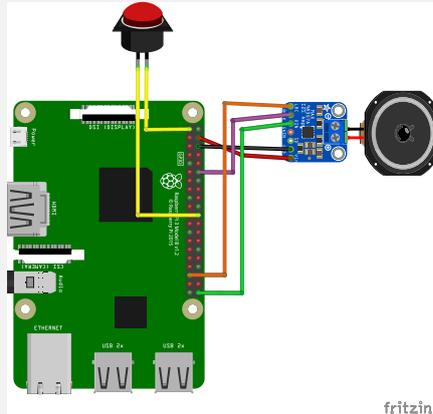
AUFBAU UND KOSTEN

Benötigte Komponenten:

Komponente	Preis
Raspberry Pi 3 B	32 €
Micro SD Card 4GB	4 €
Adafruit MAX98357A Verstärker	9 €
Lautsprecher	6 €
Taster	0,5 €
Netzteil 5 Volt	10 €
Summe:	61,50 €

Die Komponenten können beliebig in einen alten Wecker eingebaut, oder unabhängig genutzt werden.

Hardwareaufbau als Zeichnung:



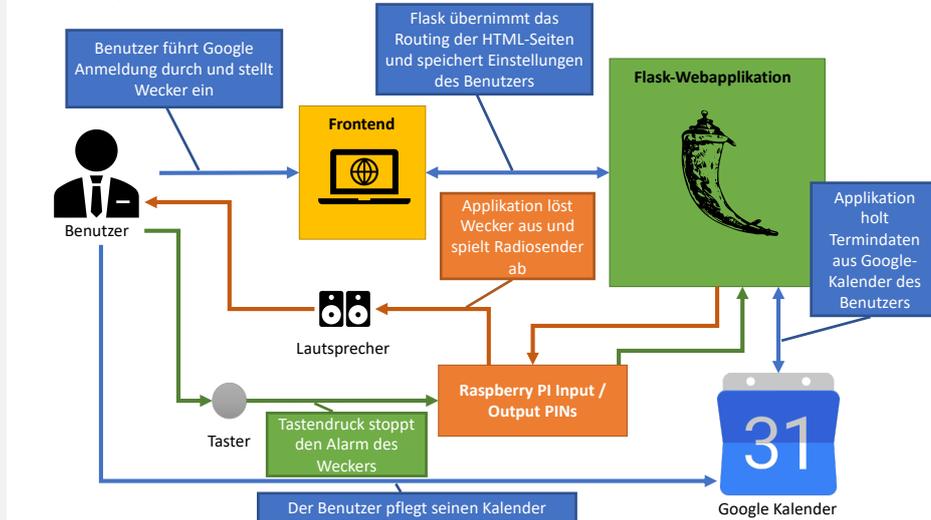
Pinbelegung:

PIN	angeschlossen an:
3.3 Volt	Taster
GPIO 25	Taster
5 Volt	Amp Vin
GND	Amp GND
GPIO 21	Amp DIN
GPIO 18	Amp BCLK
GPIO 19	Amp LRCLK

Wie links auf der Zeichnung zu sehen, müssen die einzelnen Komponenten mithilfe von Kabeln an die vorgegebenen Pins des Raspberry Pis angeschlossen werden.

FUNKTION UND BEDIENUNG DES WECKERS

Bedienungsablauf:



Beispiel-Programmablauf:



Der Benutzer ...
 .. meldet sich mit Google an.
 .. konfiguriert den Wecker.
 .. pflegt seinen Terminkalender fortlaufend.
 .. wird rechtzeitig zum Termin geweckt.

ERWEITERUNGSMÖGLICHKEITEN DES SMARTEN WECKERS

weitere Upcycling-Möglichkeiten:

- Wake-Up-Light
- Wetter- und Terminanzeige
- mehrere Kalenderdienste
- Statistiken
- Smart-Home-Funktionen
- Sprachsteuerung

REFERENZEN

Github-Page: github.com/malewric/raspimartalarmclock
Flask Dokumentation: <http://flask.pocoo.org/docs/1.0/>
Python Dokumentation: <https://docs.python.org/3/>
Raspberry Dokumentation: <https://www.raspberrypi.org/documentation/>
Google-Calendar-API Doku: <https://developers.google.com/calendar/>
Bootstrap: <https://getbootstrap.com/docs/4.0/getting-started/contents/>

Proseminar - Aus alt mach neu: Upcycling mit Raspberry Pi und Co.

Janek de Buhr
B.Sc. Wirtschaftsinformatik
Universität Oldenburg
Oldenburg

Abstract

In diesem Projekt ist es das Ziel, einen alten passiven Lautsprecher mithilfe eines Raspberry Pi zu einem innovativen neuen aktiven Lautsprecher umzufunktionieren. Am Ende des Projekts soll ein neuer Lautsprecher ein ansprechendem Design, die Alexa- und Media-Player Funktion inklusive Bluetooth- und WLAN-Steuerung beinhalten. Dazu soll ein Taster angebracht werden, der das Herunterfahren ermöglicht.

1. Einleitung

1.1. Einführung ins Seminar

Das Seminar "Aus alt mach neu: Upcycling mit Raspberry Pi und Co." besteht aus der praktischen Umsetzung eines alten Gerätes, welches zu einem neuen umfunktioniert werden soll und der theoretischen Umsetzung, die eine Ausarbeitung eines IEEE-Papers, Posters, Handbuchs und der Präsentation beinhaltet. Zuallererst geht es darum, sich Gedanken über ein altes Gerät zu machen, was upgecycelt werden kann. Wenn man sich auf ein Gerät festgelegt hat, soll man Ideen zum upcyclen herausarbeiten und im Nachhinein entscheiden, welche Ideen in Angriff genommen und umgesetzt werden. Es gibt in gewissen Abständen, ähnlich wie Meilensteine, ein Treffen mit dem Betreuer und stellt seinen Zwischenstand vor. Durch ein Feedback kann es gegebenenfalls zu Änderungen kommen.

1.2. Geschichte des Gerätes

Die Geschichte der Lautsprecher begann schon im Jahre 1860 und war damals direkt mit der Erfindung des Telefons verknüpft. Die damals entwickelten Schallwandler, erfunden von Philipp Reis und Alexander Graham Bell, dienten den Telefonen. Im Jahre 1925 erfolgte dann die Markteinführung der ersten leistungsstarken Röhrenverstärker mit einem Watt oder mehr. Im selben Jahr wur-

de auch der dynamische Tauchspulenlautsprecher von der amerikanischen Firma Western Electric entwickelt, dieser wird bis heute in den meisten schallabstrahlenden Systemen eingesetzt. Der Engländer deutscher Abstammung Paul G. A. H. Voigt zählt zu den Pionieren der Lautsprecher, er präsentierte im Jahr 1936 den ersten Lautsprecher mit "Permanent Magnet". Durch die Arbeiten des Australiers Albert Neville Thiele und des US-Amerikaners Richard H. Small im Jahre 1951 begann die Ära des modernen Lautsprechers, sie ermöglichten, dass relativ kleine Lautsprecherboxen heute erstaunlich tiefe Frequenzen abstrahlen können.

1.3. Gebrauch des Gerätes

Die eigentliche Aufgabe eines Lautsprechers ist es, den Ton wiederzugeben und das in verschiedenen Möglichkeiten.

Die Möglichkeiten, einen Lautsprecher einzusetzen und somit zu gebrauchen sind,

- Musik einer Anlage
- den Ton eines Gesprächs, z.B. von einem Telefonat,
- Klang eines spielbaren Gerätes (Gitarre)

wiedergeben.

2. Hauptteil

2.1. Ideenfindung

Wenn man einen alten Lautsprecher mit denen von heute vergleicht, wird man sehr schnell nach Upcyclingmöglichkeiten aufmerksam. Die alten Lautsprecher werden in der Regel an eine Anlage angeschlossen, dadurch mit Strom versorgt und spielen die Musik ab, die an der Anlage eingestellt wird. So ein Lautsprecher kann aber auch ganz anders und viel einfacher betrieben werden. Eine Möglichkeit wäre, die Box über **Bluetooth** mit einem Handy oder ein

ähnliches heute sehr brauchbares Gerät zu betreiben. Des Weiteren gäbe es die Möglichkeit, die Box über **WLAN** anzusteuern, um beispielsweise **Internetradio** zu hören oder einfach von jedem Gerät, was mit dem Netzwerk verbunden ist, seine Musik abzuspielen. Wenn man sich die neuen Funktionen eines Lautsprechers anschaut, stößt man automatisch auf die virtuellen Assistenten, wie zum Beispiel "Alexa". Mit der **Alexa-Funktion** ist es dann möglich, mit dem Gerät zu kommunizieren und sich viele wichtige Informationen einzuholen. Eine weitere Idee wäre, dem Lautsprecher ein ansprechendes **Display** zu verpassen, wo man angezeigt bekommt was der Lautsprecher gerade abspielt oder welche Funktion benutzt wird. Um ein schickes Design an der Box anzubringen, kann man **LED-Streifen** verwenden, um die Box mit Beleuchtung nach außen hin ansprechend zu gestalten.

2.2. Ideenauswahl

Durch die Betrachtung neuerer Lautsprecher kommt man an den oben genannten Ideen zum Upcycling nicht mehr vorbei. Die ausgewählten Ideen im Projekt sind dadurch eher klein gehalten, aber es sind die wichtigsten Merkmale eines Lautsprechers in der heutigen Zeit. Es wurde also die Umsetzung eines **Media-Player** inklusive der **Bluetooth-** und **WLAN-Ansteuerung** gezielt in Betracht gezogen. Mit der **WLAN-Funktion** kann man das Abspielen eines Internetradios ermöglichen oder sie einfach von einem beliebigen Netzwerkgerät per IP-Adresse im Netzwerk ansteuern und mittels einem Media-Server die eigene Musik abspielen lassen. Des Weiteren wurde als Extra die **Alexa-Funktion** zur Umsetzung angepeilt, da die Sprachsteuerung in der heutigen Zeit und auch in Zukunft immer wichtiger werden wird. Jeder Mensch will sich so wenig Arbeit wie möglich machen, dadurch kann er einfach im Sitzen die Box mit Sprache bedienen, ohne einen Aufwand betreiben zu müssen. Durch Alexa wird dem Menschen sehr in seinem Lebensablauf geholfen. In folgenden Punkten hilft Alexa dem Menschen:

- Wecker zum Aufstehen oder Timer zum Kochen stellen lassen.
- Verkehr, um die aktuelle Verkehrslage in Erfahrung zu bringen.
- Smart Home, um die Hausbeleuchtung darüber zu steuern.

2.3. Realisierung

Um die Upcyclingideen zu realisieren, musste zunächst ein idealer passiver Lautsprecher gefunden und so umgebaut werden, dass er mit einem Raspberry Pi bedient wer-

den kann.

Zu diesem Umbau gehörten folgende Anschaffungen:

- USB-Mikrofon
- HiFiBerry AMP2 Verstärker
- HiFiBerry AMP2 Netzteil



(a) HifiBerry AMP2

(b) USB-Mikrofon

Abbildung 1. Anschaffungen für den Umbau

Der Umbau der Box gestaltete sich folgendermaßen. Am Anfang lag ein ganz normaler passiver Lautsprecher vor, der mittels einem Raspberry Pi und HiFiBerry AMP2 Verstärker zu einem aktiven Lautsprecher umfunktioniert werden musste. Den Verstärker kann man durch die GPIO-Pins auf den Raspberry Pi stecken, um eine Verbindung zur Box herzustellen. Jetzt konnte damit begonnen werden, die Box zusammenzubauen und das Raspberry-Kit an der hinteren Seite der Box zu befestigen. Während des Zusammenbaus wurde als Extra ein Taster zum Herunterfahren eingebaut, der über die GPIO-Pins und einem Python-Programm realisiert wurde. Für die vorhergesehene Alexa-Funktion wurde ein USB-Mikrofon in der Box eingearbeitet. Nachdem alles zusammengebaut war, konnte mit der Softwarebasierten-Umsetzung des Raspberry Pi angefangen werden. Zunächst wurde ein Berryboot-Image auf die SD-Karte gespielt, um mehrere Images auf den Raspberry laden zu können. Die Funktionen "Alexa" und der "Media-Player" wurden getrennt voneinander als eigenes Image angelegt. Der Media-Player wurde mit einem Max2Play-Image realisiert. Das Image basiert auf dem Raspbian-Stretch-Betriebssystem. Mit dieser Grundlage kann man die Umsetzung von Bluetooth, WLAN und das Internetradio realisieren. Sobald das Image installiert und gebootet wurde, muss die config-Datei so angepasst werden, dass die Onboard-Soundkarten entfernt werden und die HiFiBerry hinzugefügt wird, da es sonst beim Booten zu Problemen führen kann. Wenn nach dem Booten die Soundkarte erfolgreich erkannt wurde, konnte mit der Einrichtung der Funktionen begonnen werden. Die Einrichtung und Konfiguration wurde über die Weboberfläche (siehe Abbildung 3.) von Max2Play verwirklicht, die kann man über <http://janeksbox>

oder der IP-Adresse erreichen. Sobald man die Weboberfläche geöffnet hat, wurde dort die Einstellung der Soundkarte vorgenommen. In diesem Fall musste die HiFiBerry AMP2 ausgewählt werden. Nachdem diese ordnungsgemäß eingestellt wurde, mussten die verschiedenen Dienste, wie Bluetooth und der Squeezebox Server, installiert und eingerichtet werden. Der Squeezebox-Server ist ein Logitech Media Server, hierüber wird die WLAN-Ansteuerung oder das Abspielen eines Internetradios umgesetzt. Damit der Media Server einen Audioplayer erkennt, muss im Reiter Audioplayer der Squeezeplayer eingerichtet werden, sobald dieser funktionstüchtig ist, kann der Media Player z.B. das eingestellte Internetradio abspielen. Ein weiteres kleines Feature ist die Funktion "Spotify-Connect", hier wird die Box in Spotify, falls das Gerät im selben Netzwerk ist, als Ausgabegerät angezeigt.



Abbildung 2. Max2Play Navigation

2.4. Test und Diskussion

Die Funktionalitäten wurden durch verschiedene Szenarien getestet. Um das Abspielen über Bluetooth zu testen, musste zuerst das Gerät in der Weboberfläche eingerichtet werden. Nachdem das Gerät erfolgreich gekoppelt wurde, konnte mit der Wiedergabe begonnen werden. Die Wiedergabe der Musik oder ähnlichem war erfolgreich. Die Verbindung mit der Box über Bluetooth funktioniert nach einmaliger Einrichtung einwandfrei. Um den Media-Player testen zu können, gibt man in einem Browser oder per App die IP-Adresse oder <http://janeksbox/> und den Port ein. Dadurch erreicht man die Weboberfläche des Logitech-Media-Servers und kann hier seine eigene Musik oder ein Internetradio abspielen lassen. Wichtig hierbei ist, dass der Squeezeplayer richtig eingestellt ist, weil sonst kein Player erkannt und ausgewählt werden kann. Die Funktion des Media-Player war ebenfalls erfolgreich. Die Alexa-Funktion konnte leider nicht umgesetzt werden, da immer wieder Probleme bei der Installation der "Alexa-Device-SDK" auftraten und leider war die Umsetzung nur durch diese SDK möglich. Im letzten Jahr gab es die Alexa-Sample-App, womit die Umsetzung um einiges leichter war.

3. Fazit

Im gesamten war dieses Seminar/Projekt sehr wissenswert und informativ, da das eigene Wissen durch den in-

tensiven Umgang mit einem Raspberry Pi oder der Anwendungen sehr gesteigert wurde. Die lang andauernde Fehlersuche, falls etwas nicht funktionierte, führte dazu, dass man sich sehr tief gehend in die Ordnerstruktur und Dateien eines Raspberry Pi's einarbeiten musste. Der Umbau der Box war ebenfalls sehr anspruchsvoll, zum einen das Löten der Verbindungen vom Verstärker zum Lautsprecher und zum anderen die Einarbeitung des USB-Mikrofons und Tasters. Dieses Seminar ist sehr empfehlenswert, da man sehr viel Eigeninitiative hineinstecken muss und einen sehr hohen Lerneffekt erreicht.



Abbildung 3. Lautsprecher

Literatur

- [1] <https://www.rasppishop.de/AMP2>
- [2] <https://www.connect.de/ratgeber/lautsprecher-geschichte-entwicklung-historie-3195199.html>
- [3] <http://www.hornlautsprecher.de/kapitel101.htm>
- [4] <https://www.max2play.com/en/wiki/>

Meine Lautsprecherbox

Janek de Buhr

EINFÜHRUNG IN MEIN PROJEKT

Was ist das Ziel dieses Projekts?

1. Aus einem alten Gegenstand ein neuwertiges zu machen.
2. Das Interagieren mit einem Minicomputer, wie der Raspberry Pi.
3. Eigenständige Arbeit.
4. Sich mit dem Begriff Upcycling auseinanderzusetzen und daraus Ideen zu entwickeln.

Neue Funktionen meiner Lautsprecherbox:

- Media-Player inkl. Internetradio
- Bluetooth und WLAN
- Spotify-Connect
- Taster zum Herunterfahren

Lautsprecher vorne



ALLES ÜBER DEN LAUTSPRECHER

Geschichte

- 1860: Wurden die ersten Schallwandler erschaffen, die den Telefonen dienten.
- 1925: Markteinführung der ersten leistungsstarken Röhrenverstärker mit einem oder mehr Watt.
- 1936: Wurde der erste Lautsprecher mit Permanent Magnet vorgestellt.
- 1951: Beginn die Ära des modernen Lautsprechers.

Hardwarekomponenten

- Raspberry Pi 3
- HiFiBerry AMP2 Verstärker (siehe Abbildung rechts)
- HiFiBerry Netzteil



Media-Player:

- Wurde als Logitech Media Server angelegt.
- Erreicht man unter <http://janeksbox:9000> oder der IP-Adresse:9000.
- In der Weboberfläche kann man den Radiosender einstellen oder eigene Musik abspielen.

Bluetooth:

- Wurde als eigenständiger Dienst eingerichtet.
- Gerät muss zuerst mit der Box gekoppelt werden und kann dann die Musik abspielen.

Spotify-Connect:

- Ist ebenso wie Bluetooth ein eigenständiger Dienst.
- Hierbei gibt man seine Login-Daten an und somit erscheint, in der eigenen App, die Box als Ausgabegerät.

- **Wichtig: Es muss ein Premium-Account vorhanden sein!**

Taster:

- Dient zum Herunterfahren, wurde über die GPIO-Pins und einem Python-Programm realisiert.

BILDER VOM LAUTSPRECHER

Lautsprecher hinten



Raspberry Gehäuse innen



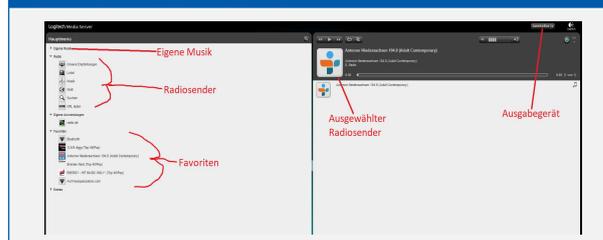
Taster zum Herunterfahren



KOSTENÜBERSICHT

- Raspberry Pi 3 - 32,90€
 - SD-Karte 16GB - 7,99€
 - HiFiBerry AMP2 - 49,99€
 - HiFiBerry Netzteil - 20,00€
 - Lautsprecher - 20,00€
- Gesamtpreis: 130,88€

LOGITECH MEDIA SERVER



LITERATURANGABEN

- <https://www.connect.de/ratgeber/lautsprecher-geschichte-entwicklung-historie-3195199.html>
- <http://www.hornlautsprecher.de/kapitel101.htm>

Aus alt mach neu: Upcycling einer Senseo Kaffeemaschine mit dem Raspberry Pi

Arne Schlömer

Aus alt mach neu: Upcycling mit Raspberry Pi und Co.

Department für Informatik

Carl von Ossietzky Universität Oldenburg

Abstract

Abfall, Plastik, Elektroschrott und alte Gegenstände: Immer mehr wird einfach Weggeschmissen. Selbst wenn die Gegenstände ihren Nutzen noch nicht komplett verloren haben und mit ein wenig Aufwand wieder zu neuem Nutzen kommen könnten. Upcycling zielt genau auf die Aufbereitung und Wiederverwendung von alten Gegenständen mit teilweise sogar komplett neuen Funktionen ab. Im Verlauf dieses Papers wird auf das Upcycling und dem damit verbundenen Umbau einer Philips-Senseo-Kaffeemaschine eingegangen.



Abbildung 1. Philips Senseo HD 7810

1. Einleitung

Eine Kaffeemaschine ist in nahezu jedem Haushalt und an fast jeder Arbeitsstätte zu finden. Durch die einfachste Form einer Kaffeemaschine, der Filterkaffeemaschine, wird aus gemahlten Kaffeebohnen, heißem Wasser durch einen Aufguss und anschließender Filtrierung Kaffee hergestellt. Je nach Typ der Kaffeemaschine ist es möglich, auch andere Heißgetränke zuzubereiten. Beim Upcycling geht es darum, Gegenstände, die ihren Nutzen noch nicht komplett verloren haben, vor dem Wegschmeißen zu bewahren. Dies gelingt durch Aufbereitung und Wiederverwendung der alten Gegenstände. Teilweise werden den veralteten Gegenständen neue Funktionen und Anwendungsbereiche hinzugefügt. Dadurch wird durch die Wiederverwendung der Gegenstände und Materialien, die Verwendung von Rohstoffen reduziert. Im Folgenden wird eine alte Philips-Senseo-Kaffeemaschine mithilfe eines Einplatinencomputers upgecycled.

1.1. Gebrauch der Philips-Senseo-Kaffeemaschine

Die Philips-Senseo-Kaffeemaschine ist eine sogenannte abgewandelte Kaffeemaschine, die mithilfe von Kaffeepads und Wasser per Knopfdruck tassenweise Kaffee und

andere Heißgetränke zubereiten kann. Dabei wird das vorher in den Wassertank gefüllte Wasser nach dem Einschalten der Senseo-Kaffeemaschine innerhalb von wenigen Minuten auf ca. 90°C erhitzt. Das Kaffeepulver wird in Form von Kaffeepads in das Gerät eingelegt. Ein Kaffeepad enthält mit etwa 7g, die für eine Tasse Kaffee empfohlene Menge Kaffeepulver. Mit einem Druck von etwa 1,4 Bar wird das Wasser durch das Kaffeepad gedrückt. Durch die kurze Kontaktzeit bleiben die Reiz- und Bitterstoffe im Kaffeepad zurück. Je nach Wunsch können entweder ein Pad für eine Tasse Kaffee oder zwei Pads für zwei Tassen frischen Kaffee verwendet werden. Des Weiteren verfügt die Philips-Senseo-Kaffeemaschine über eine automatische Abschaltfunktion nach 60 Minuten.

1.2. Einschränkungen

Im Vergleich zu heutigen Kaffeemaschinen fehlen der Philips-Senseo HD 7810 einige Funktionen. Es ist zum Beispiel nicht möglich, eine ganze Kanne Kaffee zu kochen um mehrer Personen mit Kaffee zu versorgen. Desweiteren ist es im Vergleich zu den neusten Modellen von Kaffeemaschinen nicht möglich, die Kaffeemaschine in das

„Smart-Home“ Heimnetzwerk einzubinden und zu steuern. Die Philips-Senseo-Kaffeemaschine verfügt zwar über eine automatische Abschaltfunktion nach 60 Minuten, jedoch wäre es effizienter die Kaffeemaschine nach einer geringeren Zeit abzuschalten. Eine Funktion um eine Uhrzeit für das automatische Einschalten der Maschine festzulegen, würde weiteren Komfort bieten. Ebenso eine bessere Darstellung des aktuellen Status der Kaffeemaschine.

2. Ideen zum Upcycling und neue Funktionalitäten

Durch die Erweiterung der Senseo-Kaffeemaschine mit einem Einplatinencomputer sollen neue Möglichkeiten, Funktionen und Verwendungen für die Senseo-Kaffeemaschine ermöglicht werden. Dadurch wird der wegen mangelnder Funktionalität oder Verwendung oftmals zu schnell weggeworfenen Senseo Maschine, eine neue Verwendung gegeben. Es soll möglich sein, die Philips-Senseo-Kaffeemaschine über eine Weboberfläche von jedem Endgerät mit Browser und Heimnetzzugang zu steuern. Die bisherigen Funktionen sollen um die Möglichkeit einer zeitgesteuerten Einschaltfunktion erweitert werden. Desweiteren soll der Status der Philips-Senseo-Kaffeemaschine in der Weboberfläche ersichtlich sein oder stärker visualisiert werden. Die Senseo-Kaffeemaschine sollte energieeffizient handeln, indem sie sich nach einer gewissen Zeit oder nach dem Kochen einer Tasse Kaffee abschaltet, sofern es vorher gewünscht war. Außerdem sollte sie sich abschalten, sobald kein Nutzer und somit kein Netzwerk mehr in Reichweite ist.

2.1. Verbesserungen

Durch eine Ansteuerung der Senseo Maschine über den Raspberry Pi können Funktionen der Kaffeemaschine z.B. auf einer Weboberfläche oder in einer App visualisiert werden. Weiterhin kann durch Verwendung einer LED-Leuchte und dem Auslesen des Status der Maschine dieser visualisiert werden. Durch eine Timer-Funktion kann die Philips-Senseo-Kaffeemaschine sich zu einer vorher eingestellten Zeit automatisch einschalten. Das Ganze sollte über einen Einplatinencomputer realisiert werden.

3. Planung

3.1. Hardwarearchitektur

Benötigte Hardware:

- Philips-Senseo HD7810 Kaffeemaschine
Die Philips-Senseo-Kaffeemaschine bietet die Grundlage für das Upcycling-Projekt. Mit ihr wird der Einplatinencomputer verbunden werden.

- RaspberryPi Zero
Der Raspberry Pi Zero mit einem onboard Wireless LAN Chip und einer GPIO (General Purpose Input Output - Allzeckeingabe/-ausgabe) 40 Pin Leiste bietet die Grundlage zur Steuerung der Kaffeemaschine. Auf ihm werden mithilfe eines Frameworks und einem Pythonprogramm die GPIO-Pins gesteuert. Er bietet die Möglichkeit über das Wlanetzwerk mit den Endgeräte zu kommunizieren und Befehle an die Philips-Senseo-Kaffeemaschine weiterzuleiten. Der Raspberry Pi Zero W 1.1 verfügt über einen 1 GHz ein Kern Prozessor, 512MB RAM, einen Mini HDMI Port, einen Micro USB OTG Port, einen Micro USB Power Port, composite video and reset headers und einen CSI camera connector. Zusätzlich wird ein Micro USB Kabel mit 2A Stromadapter und eine Micro-SD Karte mit 4 GB benötigt.
- 5V Relay mit Optokoppler
Um den Status der Philips-Senseo-Kaffeemaschine über die GPIO-Pins auslesen zu können und dabei dem Raspberry Pi unter Umständen nicht der Spannung der Philips-Senseo-Kaffeemaschine von 240 Volt auszusetzen, muss ein Relay zur Trennung des Stromkreises verwendet werden. Die GPIO-Pins des Raspberry Pi sind nur für eine Spannung von 3.3 Volt ausgelegt.
- Kabel
Für die Verbindung des Raspberry Pi mit der Senseo-Maschine werden Jumper Kabel verwendet, welche zum verlöten an der „male“-Seite abgeschnitten und abisoliert werden.
- NeoPixel Ring
Zur Visualisierung des Status wird ein NeoPixel Ring 16 LED RGB verwendet, welcher durch den 5 Volt Ausgang des Raspberry Pi betrieben werden kann.

3.2. Softwarearchitektur

Der Raspberry Pi Zero wird mit dem Betriebssystem „Rasbian Stretch with Desktop“ betrieben. Auf dem Betriebssystem wird das Python Framework WebiOPi als Service betrieben. Webiopi stellt einen Webserver zur Verfügung, mit dem es möglich ist, über eine HTML-Seite mittels HTML Komponenten und Javascript mit Pythonprogrammen zu kommunizieren. Die HTML Seite, sowie auch die Python Datei ist komplett über frei konfigurierbar und austauschbar. Über die angesteuerten Pythonprogramme können die GPIO-PINS des Raspberry Pi gesteuert werden und Input Werte verarbeitet werden. Die Funktionsweise ist beispielhaft in Abbildung 2. dargestellt. Hierbei wird zunächst die HTML-Datei angefordert und daraufhin in der HTML-Datei die Javascript-Datei. Weiterhin

wird im „Script-Bereich“ die `webiopi().setup(..)`-Funktion mit Parametern aufgerufen, wodurch der WebiOPi-Service die `setup`-Funktion im Pythonprogramm ausführt. Durch das Anklicken eines Buttons auf der HTML-Seite wird eine Macro-Call-Funktion mit der aufzurufenden Methode aufgerufen, welche dazu führt, dass der WebiOPi-Service die im Pythonprogramm definierte Methode aufgerufen wird und wenn sie einen Rückgabewert enthält, dieser an die HTML-Seite zurück gegeben wird.

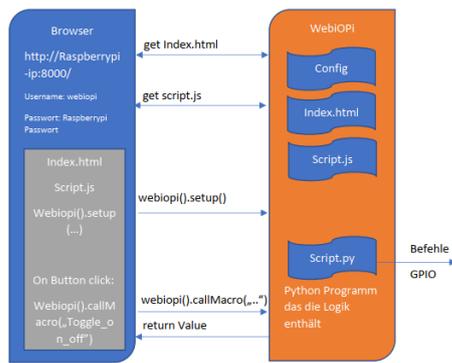


Abbildung 2. WebiOPi Funktionsweise

4. Realisierung

Um das Upcycling umzusetzen, muss zunächst die Philips-Senseo-Kaffeemaschine auseinander gebaut werden, damit daraufhin alle Komponenten miteinander verbunden werden können. Anschließend kann das WebiOPi-Framework installiert und die Programme und Dateien können angepasst werden. Als letztes muss die Verkabelung überprüft und alle Funktionen getestet werden.

4.1. Verkabelung der Hardware

Um den Raspberry Pi mit der Philips-Senseo-Kaffeemaschine zu verbinden, müssen die Kabel mit der Platine der Senseo-Kaffeemaschine verlötet werden. Dazu muss die Philips-Senseo-Kaffeemaschine komplett zerlegt werden, sodass die Platine herausgenommen werden kann. Bei dem Ausbau der einzelnen Komponenten der Philips-Senseo-Kaffeemaschine ist drauf zu achten, dass sich der Netzwerkstecker nicht in der Steckdose befindet. Für die Umsetzung der neuen Funktionen werden je ein Kabel für das Einschalten, der Zubereitung von einer Tasse, der Zubereitung von zwei Tassen, das Auslesen der Status Led und ein Kabel für das Verbinden der Masse benötigt. Um mit der Led experimentieren, zu können wurde ein weiteres Massekabel vor der Led verlötet. Um der Senseo ein Signal zu geben, werden die Kabel für das Einschalten

sowie für das Zubereiten vor die Taster gelötet. Dies sorgt dafür, dass sobald ein Output-Signal vom Raspberry Pi an die Senseo-Kaffeemaschine gesendet wird, diese denkt das der Taste ausgeöst worden ist und das Signal an den auf der Senseo-Platine verbauten Controller weiterleitet, welcher die Funktion auslöst.



Abbildung 3. Verlötete Platine der Philips-Senseo-Kaffeemaschine

Anschließend müssen die Kabel mit dem Raspberry Pi verbunden werden. Dazu werden die Kabel für das Einschalten, Masse, eine Tasse und zwei Tassen direkt auf die Raspberry Pi GPIO-Pins gesteckt. Die Led wird mit dem 5V Relay verbunden, welches wiederum mit dem Raspberry Pi verbunden werden muss. Um den Status visuell darstellen zu können muss noch der NeoPixel-Ring mit dem Raspberry Pi verbunden werden. Die genaue Verkabelung kann Abbildung 4. entnommen werden.

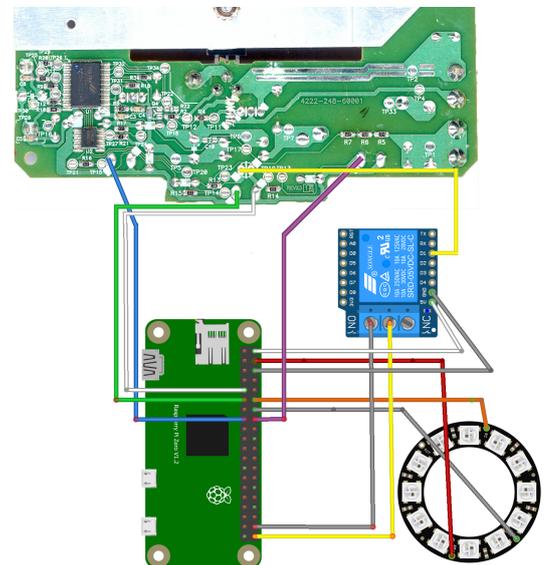


Abbildung 4. Verkabelung des Raspberry Pi mit der Philips-Senseo-Kaffeemaschine

4.2. Softwarer Anpassungen

Das Rasbian Betriebssystem kann von der Raspberry.org-Seite heruntergeladen und auf einer Micro-SD installiert werden. Nach erfolgreicher Installation muss das WLAN eingerichtet werden, dies kann einerseits in den Konfigurationsdateien erfolgen oder über einen Bildschirm mithilfe von Tastatur und Maus eingerichtet werden. Um den WebiOPi-Service nutzen zu können, muss dieser über das Terminal mit einem Befehl installiert werden. Wenn der Service erfolgreich installiert ist kann ein Ordner mit einem beliebigen Namen angelegt werden. In diesem müssen zwei Unterordner mit den Bezeichnungen html und python erstellt werden. In dem html-Ordner wird eine index.html-Datei und im python-Ordner die script.py-Datei erstellt. Anschließend muss die config-Datei von WebiOPi abgeändert werden, sodass WebiOPi die neu angelegten Dateien verwendet anstelle der Standard-Dateien. In der index.html-Datei werden die Elemente hinzugefügt, um die Philips-Senseo-Kaffeemaschine über die Weboberfläche zu steuern. Dazu gehören der Power On/Off-Button, eine Tasse-Button, zwei Tassen-Button, ein Eingabefeld zum Einstellen der Timer Uhrzeit, einen Senden-Button zum Absenden der Uhrzeit, einen Button um den Timer aus- oder anzustellen sowie Bilder zur Statusanzeige der Maschine. Allen Buttons werden Funktionen zugewiesen, die bei einem Klick auf die Buttons ausgeführt werden und gegeben Falls per `webiopi().callMacro()`-Befehl eine Pythonmethode aufrufen. Für die Statusanzeige wurden vier Bilder hinzugefügt (Wasser leer, Aufheizen, Aus, An, Timer an), welche je nach Status aus- oder eingeblendet werden. Die Status Abfrage wird durch eine im html-Script definierte loop-Funktion alle zwei Sekunden überprüft. Im Pythonprogramm wurde eine Setup Methode definiert, welche beim Start des Services ausgeführt wird. Weiterhin sind Methoden definiert, die die Funktionen der Buttons abdecken und den Status der Philips-Senseo-Kaffeemaschine überprüfen, Rückgabewerte liefern oder diese zwischenspeichern. Ein Teil der Methoden zur Statusüberprüfung ist es, den NeoPixel-Ring zu steuern und die Led Farben zu setzen bzw. sie auszuschalten, falls die Maschine aus ist. Hierbei steht die Farbe Grün für den Status an/bereit, die Farbe Rot dafür, dass die Kaffeemaschine aufheizt und die Farbe Blau dafür, dass das Wasser leer ist. Beim Beenden des WebiOPi Services wird die Philips-Senseo-Kaffeemaschine und der NeoPixel-Ring ausgeschaltet. Eine im Pythonprogramm definierte Loop-Funktion überprüft dauerhaft den Status der Maschine und vergleicht die gestellte Timeruhrzeit mit der aktuellen Uhrzeit um bei Übereinstimmung die Maschine einzuschalten.

Senseo-Wifi

Bedienung durch klicken auf die Bilder.

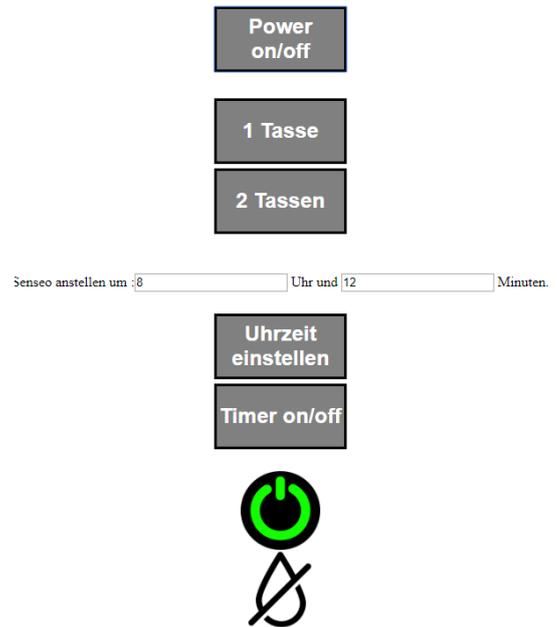


Abbildung 5. Weboberfläche zur Bedienung der Philips-Senseo-Kaffeemaschine

4.3. Test

Der Raspberry Pi bootet ohne Probleme und startet den Service durch die Autostart Funktion selbstständig. Akustisch und auch visuell ist die durch ein kurzes klackern und blinken der Led zu erkennen. Die Weboberfläche lässt sich über die IP-Adresse des Raspberry Pi und den Port 8000 aufrufen. Der Benutzername und das Passwort werden angenommen. Der Power On Button funktioniert. Die GUI zeigt an, dass die Senseo eingeschaltet worden ist, dass das Wasser leer ist und der NeoPixel-Ring verändert seine Farbe auf Blau, da der Tank nicht eingesetzt ist. Nach einsetzen des mit Wasser gefüllten Tanks fängt die Senseo an aufzuheizen. Das Bild für fehlendes Wasser verschwindet, die Farbe des NeoPixel-Rings ändert sich auf Rot und das Bild für das Aufheizen wird angezeigt. Das Stellen des Timers auf die Uhrzeit in zwei Minuten klappt auch. Das Timer Bild wird angezeigt. Nun wird der usknopf bwätigt. Die Senseo-Kaffeemaschine und NeoPixel-Ring gehen aus. Nach kurzer Zeit schaltet sich die Maschine automatisch an. Nach dem drücken des eine Tasse Buttons fängt die Senseo an eine Tasse Kaffee zu kochen. Ebenso bei dem zwei Tassen Button. Da der gleiche Blinkrythmus, wie beim Aufheizen von der Senseo verwendet wird, wird hier der Aufhei-

zenstatus angezeigt.

5. Fazit

5.1. Erfüllungsgrad der Verbesserung

Die Bedienung über eine Weboberfläche wurde zu 100% umgesetzt. Es ist möglich, die Philips-Senseo-Kaffeemaschine von allen Endgeräten, die sich im gleichen WLAN Netzwerk wie die Philips-Senseo-Kaffeemaschine befinden, über die Weboberfläche zu bedienen. Dabei kann sie über die Oberfläche ein- und ausgeschaltet werden. Es können ein oder zwei Tassen Kaffee gekocht werden, ein Timer gestellt, aktiviert oder deaktiviert werden und der Status wird durch Bilder angezeigt. Mittels des NeoPixel-Rings findet eine Visualisierung auch außerhalb der Weboberfläche statt. Die Funktion, dass sich die Kaffeemaschine nach einer bestimmten Zeit oder nach dem Kochen einer Tasse Kaffee ausschaltet, wurde aus zeitlichen Gründen nicht umgesetzt. Ebenso das Ausschalten, wenn sich kein Endgerät mehr in der Nähe befindet.

5.2. Probleme

Für die Verlotung der Kabel mit der Platine der Philips-Senseo-Kaffeemaschine war es von Nöten, die Kabel an der richtigen Stelle anzubringen ohne die Platine zu beschädigen. Der Status der Philips-Senseo-Kaffeemaschine wird nicht genau gespeichert und muss anhand des Blinkrythmus der Led ausgelesen werden. Das Auslesen ist sehr ungenau. Um den Blinkrythmus mit dem Raspberry Pi auslesen zu können, muss sicher gestellt werden, dass die Spannung der Kaffeemaschine nicht auf die GPIO Pins des Raspberry trifft. Dann musste für jeden einzelnen Status der Rythmus bestimmt und in einer Pythonmethode festgehalten werden. Weiterhin bereitet die Kommunikation über das Framework Verzögerungen in der Kommunikation zwischen den verschiedenen Komponenten, was zu Verwirrungen bei der Bedienung führen kann.

5.3. Vision

Nach der Umsetzung der bisherigen Funktionen kann die Weboberfläche in Form einer App oder ähnlichem für Smartphones und Tablets umgesetzt werden, um es dem Entanwender komfortabler zu machen. Ergänzend ist eine Anbindung an andere Smart Home Geräte einen Ansatz, der zu Verfolgen wäre. Weiterhin sollte an dem schlichten Design der Weboberfläche gearbeitet werden, um sie attraktiver für den Endanwender zu machen. Eine Pumpe, die den Wassertank auffüllt, sobald er leer ist, würde den Komfort erhöhen, da sie die Philips-Senseo-Kaffeemaschine den

Aufheizvorgang erst dann startet, wenn auch genügend Wasser vorhanden ist.

5.4. Zusammenfassung

Im Großen und Ganzen erfüllt die Philips-Senseo-Kaffeemaschine ihren Nutzen auch ohne Upcycling, sie kann weiterhin auch ohne die Verwendung des Raspberry Pi im vollem Umfang genutzt werden. Eine Kombination mit anderen Smart Home Geräten würde dem Ganzen mehr Nutzen verleihen, da die Kaffeemaschine mit einem Wecker oder Ähnlichem verknüpft werden könnte. Der Umbau war fordernd und verlangte handwerkliches Geschick beim verlöten sowie verbinden der Komponenten.

Literatur

- [1] Alina Schadwinkel, Julian Stahnke, Julius Tröger (22.08.2018), in: *Upcycling - Dieser Müll steht dir aber gut*, URL: <https://www.zeit.de/zeit-magazin/mode-design/2018-08/upcycling-mode-plastik-muell-verarbeitung>, (Stand: 04.05.2019)
- [2] URL: <https://de.wikipedia.org/wiki/Upcycling>, (Stand: 26.09.2019)
- [3] Device Plus Editorial Team (14.03.2016), in *Raspberry Pi WebIOPi IOT Part 1 – Installation & LED Blinking – Operating GPIO With a Browser*, URL: https://www.deviceplus.com/how-tos/raspberrypi-guide/raspberrypi_entry_030/, (Stand: 26.06.2019)
- [4] URL: <https://de.wikipedia.org/wiki/Senseo>, (Stand: 26.06.2019)
- [5] *WebIOPi*, URL: http://webiopi.trouch.com/Tutorial_Basis.html, (Stand 26.06.2019)
- [6] Philip Burgess, Kattni Rembor, Iady ada , *Adafruit NeoPixel Überguide*, URL: <https://learn.adafruit.com/adafruit-neopixel-uberguide/python-circuitpython#python-installation-of-neopixel-library-17-9>, (Stand: 26.06.2019)
- [7] JohannesK (02.05.2013), *Senseo with Pi*, URL: <http://senseowithpi.blogspot.com/p/tutorial-part-1.html>, (Stand: 26.06.2019)

Upcycling einer Philips Senseo Maschine mit dem RaspberryPi

Arne Schlömer

UPCYCLING - WAS IST DAS UND WAS KANN MAN AUS EINER ALTEN SENSEO-KAFFEEMASCHINE MACHEN?

- Veraltete Senseo-Kaffeemaschine werden oftmals weggeschmissen oder ersetzt, obwohl Sie noch funktionieren. Durch das Upcycling sollen die bisherigen Funktionen erweitert aber auch erhalten bleiben:
 - Bedienung durch drei Tasten an der Maschine (Power On/Off, 1 Tasse & 2 Tassen)
 - Statusanzeige durch eine Led
- Die Erweiterung mit einem Einplatiniencomputer soll folgende neue Funktionen bringen:
 - Steuerung der "normalen" Funktionen über eine Weboberfläche (Website)
 - Anzeige des Status der Kaffeemaschine auf der Weboberfläche & Visualisierung durch Leds
 - Automatisches Einschalten durch einen vorher gestellten Timer

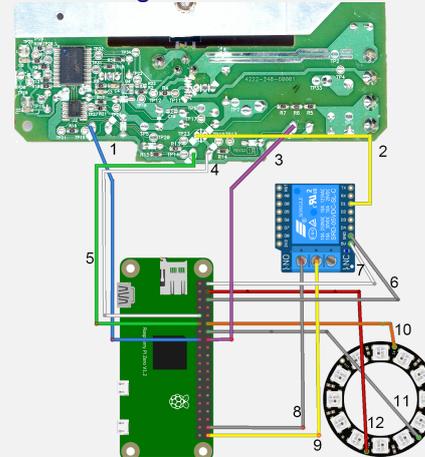


HARDWAREARCHITEKTUR

Hardwarekomponenten

- Philips Senseo HD 7810
- Raspberry Pi Zero W V1.1 mit Wlan-Modul und 40 GPIO Pins
- NeoPixel - Ring 16 RGB Leds
- 5V Relay mit Optokoppler
- Verbinden der Platine mit dem Raspberry Pi Zero
 - zwischenschalten eines 5V Relays mit Optokoppler
 - anschluss hinter den Buttons
- Visuelle Ausgabe über den NeoPixel-Ring
- Betreiben des Raspberry Pi mit einem 2A Stromadapter und Micro-USB Kabel

Verkabelung



- 1 Ansteuerung 2Tassen
- 2 Auslesen Led Status
- 3 Ansteuerung 1 Tasse
- 4 Masse
- 5 Ansteuerung Power
- 6 Masse 5V Relay
- 7 5V Stromversorgung Relay
- 8 Pi Out - Relay In
- 9 Pi In - Relay Out
- 10 Signal NeoPixel
- 11 Masse NeoPixel
- 12 5V Stromversorgung NeoPixel

SOFTWAREARCHITEKTUR

Softwarekomponenten

- Rasbian Betriebssystem
- WebiOPi Framework - Stellt Webserver und Script Verarbeitung bereit
- Angepasste HTML-Startseite
- Python Programmcode

Nach dem Installieren des Betriebssystems und dem **WebiOPi-Framework** muss die **Index.html-Seite** angepasst werden. Sie ist die Startseite, welche über einen **Webbrowser** aufgerufen werden kann. Diese kommuniziert mittels Javascript mit dem **Pythonprogrammcode**, welcher die Logik enthält.

Kommunikation zwischen den Softwarekomponenten und dem Browser

Senseo-Wifi

Bedienung durch klicken auf die Bilder:

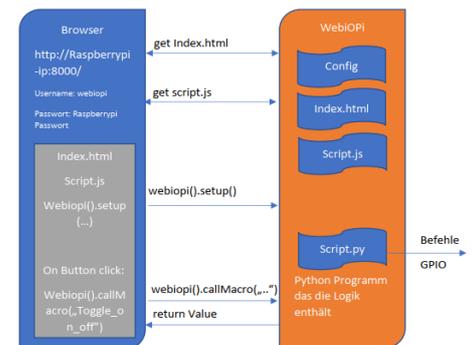


Klicken auf den Button ruft über eine JavaScript Funktion die Python Methode auf.

```

@webiopi.macro
def toggle_on_off():
    global statuspower
    GPI.digitalWrite(17, 0)
    webiopi.sleep(1)
    GPI.digitalWrite(17, 1)
    if (statuspower == 0):
        statuspower = 1
    else:
        statuspower = 0
    
```

Durch das Betätigen der Buttons auf der Weboberfläche, werden die **Methoden im Pythonprogrammcode** ausgeführt. Dieser verarbeitet die Eingabe und leitet den **Befehl an die Senseo** weiter. Weiterhin liest er über eine **Dauerschleife den Status der Kaffeemaschine** aus.



GESAMTKOSTEN

Raspberry Pi Zero	15 €
Philips Senseo Kaffeemaschine	~ 20 €
NeoPixel - Ring 16 Led	15 €
5V Relay mit Optokoppler	2 €
Kabel (Jumper Kabel - Male/Female)	4 €
Lötzinn & Heißkleber	<1 €
Gesamtkosten	57 €

LINKS & REFERENZEN

- WebiOPi - <https://webiopi.trouch.com/>
- Adafruit NeoPixel Ring - <https://www.adafruit.com/product/1463>
- Raspberry Pi WebiOPi IOT - https://www.deviceplus.com/howtos/raspberrypi-guide/raspberrypi_entry_030/
- Philips Sense-Kaffeemaschine - <https://www.philips.de/c-m-ho/kaffee/senseo-kaffeemaschine>
- Raspberry Pi - <https://www.raspberrypi.org/>

Upcycling eines Briefkastens

Anatolij Fandrich

Abstract

In dieser Ausarbeitung wird beschrieben, wie ein Briefkasten mit Hilfe des Mikrocontrollers ESP8266EX und der Anbindung an ein WiFi-Netzwerk „geupcylced“ werden kann. Der Fokus bei der Entwicklung wurde dabei auf eine möglichst lange Akkulaufzeit des Systems gelegt. Um dies zu gewährleisten wurde eine Platine entworfen, die im Gegensatz zu den verfügbaren Evaluationsboards nur die für den Betrieb nötigste Hardware enthält und die Möglichkeit besitzt den Energieverbrauch im Standby auf 0µA zu beschränken. Mittels des Kommunikationsprotokolls MQTT und dem Internet-of-Things-Framework Node-RED kann der Nutzer des Briefkastens via Twitter oder Telegram über neue Post benachrichtigt werden.

1. Einleitung

Jeder Haushalt in Deutschland verfügt über einen Briefkasten und wahrscheinlich auch über einen WiFi-Router, welcher den kabellosen Zugang zum Internet ermöglicht. In dieser Ausarbeitung wird daher die Fragestellung erläutert, wie sich diese beiden Alltagsgegenstände miteinander kombinieren lassen, um so den Funktionsumfang und die Wertigkeit eines Briefkastens im Rahmen eines Upcycling-Prozesses [11] zu erhöhen. Der Briefkasten wird mit Hilfe des Routers ein Teil des Internets und ist so in der Lage mit den entsprechenden technischen Vorkehrungen den Nutzer des Systems eigenständig via Twitter[10] oder Telegram[9] über den Empfang neuer Post zu benachrichtigen. Dadurch entfallen unnötige Wege zu einem leeren Briefkasten bzw. können so einen dringend erwartete Sendungen direkt in Empfang genommen nehmen. Somit lässt sich nicht nur der materielle Wert des Briefkastens, sondern auch die Lebensqualität des Nutzers durch diese simple Smart-Home-Anwendung erhöhen.

Die Ausarbeitung ist wie folgt gegliedert. Zunächst wird in Absatz 2 der typische Briefkasten kurz beschrieben und einige Möglichkeiten des Upcyclings eines Briefkastens in Abschnitt 2.1 dargestellt. Im Abschnitt 2.2 wird eine mögliche technische Realisierung vorgestellt, welche darauffolgend in den Abschnitten 4 und 5 näher spezifiziert wird. Die

Ausarbeitung schließt mit dem Abschnitt 6 ab, in welchem das Projekt abschließend bewertet wird, die Gesamtkosten des Upcyclings aufgeführt und weitere Möglichkeiten zur Erweiterung des Funktionsumfangs vorgeschlagen werden.

2. Der Briefkasten

Ein Briefkasten ist ein Behälter, welcher es Zustellern ermöglicht, Privatpersonen und Unternehmen Briefe und andere kompakte Sendungen zuzustellen. Je nach Wohnsituation sind verschiedene Varianten des Briefkastens verbreitet. So gibt es beispielsweise in sehr ruralen Gebieten Briefkästen, die direkt an der nächstgelegenen Straße platziert sind. Eine andere Variante ist der Briefkasten am Gebäude. Zudem gibt den Postschlitz direkt an einer Wand oder Haustür, sodass die Sendung unmittelbar im Wohnraum des Empfängers zugestellt wird. Postfächer, welche sich i.d.R. in der Nähe von Postfilialen befinden, sind eine weitere Möglichkeit Sendungen zu empfangen.

Unabhängig von der Form und dem Ort, an welchem sich der Briefkasten befindet, ist die Funktionsweise weitestgehend identisch. Mittels einer Klappe oder eines Schlitzes wird die Sendung in einen verschlossenen Behälter eingeworfen. Der Einwurfschlitz erschwert Diebstähle der privaten Sendungen und erlaubt es Privatpersonen, Briefe auch in Abwesenheit zu empfangen. Mit Hilfe eines Schlüssels kann der Behälter geöffnet und der Inhalt entnommen werden. Dies klappt problemlos und es gehört für viele Leute zur alltäglichen Routine, nach der Post im Briefkasten zu sehen. Da nicht alle Postkästen aus teilweise transparenten Material gefertigt werden, weiß man nur durch das Öffnen des Kastens sicher, ob man Sendungen empfangen hat. Aus diesem Grund werden im folgenden Abschnitt Ideen vorgestellt, wie man sich in Zukunft den Weg zu einem leeren Briefkasten sparen könnte.

2.1. Ideen zum Upcycling

Wenn via Instant-Messaging-Apps Sofortnachrichten versendet werden, bekommt der Empfänger nahezu unmittelbar eine Benachrichtigung auf das Smartphone mit dem Hinweis, dass eine Mitteilung empfangen wurde. Bei

eMail-Clients auf dem Computer benachrichtigt das Programm den Nutzer ebenfalls regelmäßig über den Empfang neuer Nachrichten. Briefkästen können diese Information leider nicht teilen. Daher könnte der Funktionsumfang des klassischen Briefkastens um eine eMail-Benachrichtigung oder den Versand von Sofortnachrichten erweitert werden.

Moderne Kraftfahrzeuge ermöglichen es mit der entsprechenden Ausstattung, das Fahrzeug ohne Nutzung des Schlüssels zu ent- und verriegeln. Als Nutzer greift man lediglich zum Türgriff des verriegelten Autos, kann direkt einsteigen und den Motor starten. So wäre es beispielsweise auch möglich, den Briefkasten mit einer ähnlichen Technologie auszustatten, um das Öffnen des Behälters ohne Schlüssel zu ermöglichen.

Eine weitere Möglichkeit des Upcyclings wäre die Anbringung eines Displays, um beispielsweise so den Familiennamen anzuzeigen. Zusätzlich könnte man den Zusteller mit einer Nachricht darauf hinweisen, das Paket an einem anderen Ort abzulegen oder sich beim Zusteller bedanken.

2.2. Der Briefkasten 2.0

Der erste Prototyp soll in der Lage sein, dem Empfänger mit Hilfe des Internets über neue Post im Briefkasten zu benachrichtigen. Dafür ist Hardware notwendig, die es ermöglicht Daten kabellos an ein Gateway, welches mit dem Internet verbunden ist, zu versenden. Eine andere Möglichkeit wäre die direkte Kommunikation mit Servern im Internet. Dies setzt beispielsweise einen Mikrocontroller mit TCP/IP Stack und einem WiFi Transceiver voraus.

Um Diebstahl oder eine Beschädigung durch Witterungseinflüsse zu verhindern, müssen die technischen Ergänzungen subtil innerhalb des Briefkastens untergebracht werden, sodass der alltägliche Gebrauch nicht behindert wird. Dies hat zur Folge (sofern es sich um einen metallischen Briefkasten handelt), dass zusätzlich eine externe Antenne notwendig ist, um eine kabellose Verbindung zum Internet aufbauen zu können.

Als Briefkasten wird ein Modell gewählt, welches i.d.R. an Gebäuden befestigt wird und über eine Klappe verfügt, die nur beim Einwurf kurz geöffnet wird. Da aus ästhetischen Gesichtspunkten auf das Verlegen von Kabeln in den Briefkasten zur Spannungsversorgung verzichtet wird, müssen sowohl Hard- als auch die Software für den Akkubetrieb optimiert sein. Zudem sollen wiederaufladbare Akkus verwendet werden und das Gesamtsystem sollte für einen möglichst langen Zeitraum wartungsfrei sein. Sobald sich die Kapazität des Akkus dem Ende neigt, muss das System den Nutzer auf einen Wechsel der Spannungsversorgung hinweisen.

Die gesamte Hardware, welche für die Funktionserweiterung notwendig ist, soll so kompakt wie möglich sein, um das Volumen innerhalb des Briefkastens nur minimal

einzuschränken. Zudem soll die Installation der Hardware unkompliziert und ohne baulichen Änderungen am Briefkasten, wie z.B. Bohrungen oder Schweißarbeiten, durchgeführt werden können.

Für den ersten Prototypen der Briefkastenelektronik ist kein ePaper Display[2] eingeplant, jedoch soll die Hardware die Voraussetzung schaffen, ein solches Display nachrüsten zu können. Dafür muss eine entsprechende Schnittstelle vorgesehen werden. Im Zuge dessen müssen auch technische Vorkehrungen getroffen werden, dem System Nachrichten für das Display zu hinterlassen. Die Kommunikation zwischen Briefkasten und Internet muss daher zumindest zeitweise bidirektional möglich sein.

3. Systemübersicht

Die folgende Abbildung 1 stellt eine Übersicht der Kommunikationsarchitektur des Gesamtsystems dar und verdeutlicht, wie eine Nachricht vom Briefkasten bis zum Smartphone des Empfängers gelangt.

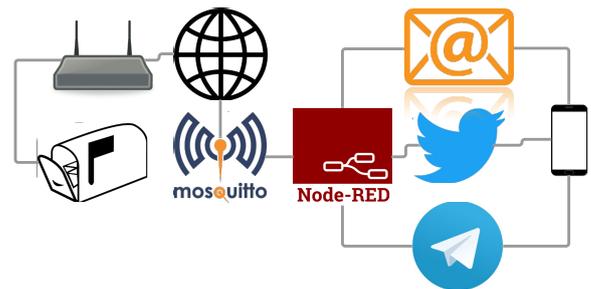


Abbildung 1. Kommunikation vom Briefkasten zum Smartphone

Die Elektronik innerhalb des Briefkastens verbindet sich beim Einwerfen einer Sendung mit einem vorher spezifizierten WiFi-Netzwerk und versendet den Akkustand über das Internet mit Hilfe der *Message Queue Telemetry Protocols* (kurz MQTT). Diese Daten werden einem MQTT-Broker (in diesem Fall Mosquitto [7]) empfangen und an einen Server, auf welchem das Internet-of-Things-Framework Node-RED [8] eingerichtet ist, weitergeleitet.

Innerhalb von Node-RED wird der Akkustand ausgewertet, eine Benachrichtigung verfasst und beispielsweise via eMail, Twitter oder Telegram an das Smartphone des Nutzers versendet. Sollte der Akkustand im Briefkasten niedrig sein, so erhält der Nutzer eine zusätzliche Meldung, die die Restspannung des Akkus enthält.

4. Hardwareentwicklung

Da sich die Hardware innerhalb des Briefkastens befinden wird, ist eine externe Antenne notwendig, um sich mit einem WiFi-Netzwerk zu verbinden. Aus diesem Grund wurde das Modul ESP-07S [5] von AI-Thinker [1] verbaut, da dieses über einen U.FL Steckverbinder verfügt. In diesem Modul befindet sich der Mikrocontroller ESP8266EX [4] aus dem Hause Espressif [3]. Der 32-Bit Mikrocontroller unterstützt die WiFi Protokolle 802.11 b/g/n, ist kostengünstig, verfügt über einen 10-Bit Analog-Digital-Konverter, welcher zur Überwachung des Akkustands verwendet werden kann, und eignet sich daher gut für eine eigene Smart-Home Anwendung.

Die Abbildung 2 zeigt die gesamte Elektronik, welche sich im Briefkasten befindet. Die Kabel sind mit Klebeband an den Innenwänden fixiert, um eine Beschädigung durch den Einwurf von Sendungen zu vermeiden.

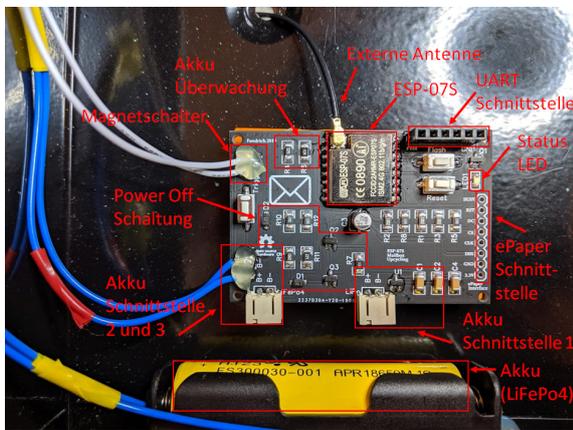


Abbildung 2. Platine, Kabel zum Magnetschalter und Akku im Briefkasten.

Die Spannungsversorgung soll durch einen Akku gewährleistet werden, welcher an einer von drei möglichen Schnittstellen mit der Platine verbunden wird. Die erste Schnittstelle verfügt über einen Spannungsregler, welcher es ermöglicht, einen Akku mit einer Maximalspannung von 6V mit der Platine zu verbinden. Denkbar wären beispielsweise LiPo-Akkus für diesen Anschluss.

Die zweite und dritte Schnittstelle sind jeweils direkt mit den Versorgungsleitungen zum Mikrocontroller verbunden. Daher dürfen nur Akkus und Batterien angeschlossen werden, welche in einem Spannungsbereich 2.5V bis 3.6V liegen. Zum Anschließen kann man zwischen einem JST-PH-Connector oder zwei freien Lötstellen wählen. Für diese Anschlüsse eignen sich zum Beispiel LiFePo4 Akkus oder zwei in Reihe geschaltete Microzellen.

Um eine möglichst lange Akkulaufzeit gewährleisten zu können, wurden mehrere technische Voraussetzungen geschaffen. So ist der Akku beispielsweise nur über eine spezielle Schaltung mit dem Mikrocontroller verbunden. Mit Hilfe des Magnetschalters, öffnet die Schaltung eine Verbindung vom Akku zum Mikrocontroller. Diese Verbindung kann im Anschluss selbstständig vom Mikrocontroller getrennt werden, sodass der Verbrauch im Leerlauf $0\mu A$ beträgt. Sollte sich der Magnetschalter nicht wieder öffnen, weil sich beispielsweise eine Tageszeitung im Briefkasten befindet, kann sich der Mikrocontroller via Software in einen Schlafmodus versetzen und so den Verbrauch von etwa 30mA im Normalbetrieb auf $20\mu A$ senken. Nebenbei überwacht der Mikrocontroller den Akkustand, um den Nutzer rechtzeitig auf einen Akkuwechsel hinzuweisen.

Die verbaute Status-LED ermöglicht es dem Entwickler bestimmte Systemzustände nachzuvollziehen. So blinkt die LED in der aktuellen Version der Firmware während des Verbindungsaufbaus zum Netzwerk und leuchtet permanent, sobald eine Verbindung aufgebaut werden konnte.

Die UART-Schnittstelle erfüllt zwei Rollen. Zum einen ermöglicht sie es Statusmeldungen vom Mikrocontroller am Computer zu empfangen. Dies ist für eine Fehlersuche und Tests besonders hilfreich. Zum anderen wird der Mikrocontroller über diese Schnittstelle programmiert.

Als künftige Ergänzung ist die Anbindung eines ePaper Displays geplant. Dafür wurde bereits eine entsprechende Schnittstelle, welche pinkompatibel zu kommerziellen ePaper Modulen ist.

5. Beschreibung der Softwareentwicklung

Die für den Briefkasten notwendige Software setzt sich aus zwei Teilen zusammen. Im Abschnitt 5.1 wird die Funktion Software, welche sich im Flash-Speicher des Mikrocontrollers befindet, beschrieben. Im Abschnitt 5.2 wird die Software auf dem Node-RED Server vorgestellt.

5.1. Firmware

Mit dem Systemstart, welcher durch den am Briefkastenschlitz befestigten Magnetschalter ausgelöst wird, hält der Mikrocontroller, mit Hilfe der in Abschnitt 4 beschriebenen Schaltung, die Verbindung zwischen Akku und Versorgungsleitung, unabhängig vom Zustand des Magnetschalters, aufrecht. Im Anschluss wird mehrmals die Batteriespannung gemessen, um die Auswirkung von Messfehlern zu reduzieren. Der Mittelwert dieser Messungen wird als Gleitkommazahl gespeichert. Danach versucht das Modul für eine bestimmte Zeiteinheit sich einem vorher definierten WiFi-Netzwerk zu verbinden. Schlagen alle Versuche fehl, so trennt der Mikrocontroller die Verbindung zum Akku, um Energie zu sparen. Sollte der Verbindungsversuch

erfolgreich gewesen sein, so wird im nächsten Schritt der MQTT-Broker kontaktiert.

Sofern der Broker nach einer bestimmten Anzahl an Fehlversuchen nicht erreichbar ist, trennt der Mikrocontroller die Verbindung zur Versorgungsspannung. Falls der Broker jedoch auf den Verbindungsversuch antwortet, versucht das Modul den Akkustand zu versenden. Analog zum vorigen Verfahren, wird nach einer bestimmten Anzahl an Fehlerversuchen das System ausgeschaltet. Sollte der Hardware-Shutdown ($0\mu\text{A}$ Verbrauch) fehlschlagen (wenn beispielsweise der Magnetschalter durch eine Zeitung immer noch geschlossen ist), wechselt das System automatisch in den Deep-Sleep-Modus ($20\mu\text{A}$ Verbrauch).

5.2. Node-RED

Innerhalb von Node-RED wird das MQTT-Topic *aamn/Briefkasten/1* vom Broker *mqtt.iot.informatik.uni-oldenburg.de* abonniert. Der Mikrocontroller im Briefkasten versendet unter diesem Topic den Akkustand. Als Schwellwert wurde 2.8V festgelegt, da der Mikrocontroller für den Systembetrieb eine Spannung von mindestens 2.6V benötigt [4]. Falls der Akkustand größer oder gleich 2.8V ist, wird eine zufällige, jedoch zuvor definierten Nachricht für den Nutzer gewählt und zeitgleich via Telegram und Twitter versendet. Sofern die Batteriespannung kleiner als 2.8V ist, so ist der Akku fast entladen und der Nutzer erhält zusätzlich den Hinweis, dass die Restspannung kritisch wird.

6. Ergebnisse der Entwicklung und Ausblick

Im Folgenden wird diskutiert, inwiefern das Upcycling des Briefkastens erfolgreich war und welche Verbesserungsmöglichkeiten für die Zukunft noch bestehen.

6.1. Ergebnisse

Der Mikrocontroller ist für dieses Projekt gut geeignet und kann den Nutzer wie gewünscht via Twitter und Telegram benachrichtigen. Durch die kompakte Platine und Kabelführung an den Innenwänden, kann der Briefkasten ohne Einschränkungen verwendet werden. Der verbaute Magnetschalter hat sich ebenfalls als praktisch und einfach zu installieren erwiesen. Der Verbrauch im Leerlauf beträgt wie gewünscht $0\mu\text{A}$. Durch die Verwendung von MQTT ist ebenfalls eine bidirektionale Kommunikation zum Internet möglich. Die ePaper-Schnittstelle funktioniert.

Die Gesamtkosten ohne Briefkasten und Verbrauchsmaterial betragen etwa 20€ .

Sämtliche Ergebnisse dieses Projekts sind in einem Git-Repository abgelegt [6]. Dort befindet sich die Firmware, der Node-RED Flow, der Schaltplan und die Gerber-Files

der Platine, eine exakte Preisübersicht, 3D-Modelle, eine Liste mit den verwendeten Technologien, weitere Abbildungen und Fotos, eine umfangreichere Dokumentation und eine Anleitung zur Inbetriebnahme.

Insgesamt kann die Entwicklung als erfolgreich bezeichnet werden, da alle in 2.2 beschriebenen Anforderungen erfüllt sind.

6.2. Ausblick

In der nächsten Version des Prototypen könnte das ePaper-Display montiert werden. Zudem wäre die Anbringung eines Keypads zum Eintippen einer PIN zum Entriegeln des Briefkastens eine interessante Ergänzung. Außerdem könnte eine Version entwickelt werden, die auf SMD-Bauteile verzichtet, um es so auch Novizen zu ermöglichen, eine solche Platinen selbst zusammenlöten zu können. Da das System mit Node-RED und MQTT äußerst modular aufgebaut ist, sind auch andere Möglichkeiten zur Benachrichtigung, wie per Beleuchtungseinheiten, umsetzbar.

Literatur

- [1] Ai-Thinker, <https://www.ai-thinker.com/home>, Letzter Zugriff: 28.06.2019.
- [2] Elektronisches Papier, https://de.wikipedia.org/wiki/Elektronisches_Papier, Letzter Zugriff: 28.06.2019.
- [3] Espressif, <https://www.espressif.com/>, Letzter Zugriff: 28.06.2019.
- [4] ESP8266EX, <https://www.espressif.com/en/products/hardware/esp8266ex/overview>, Letzter Zugriff: 28.06.2019.
- [5] ESP-07S, <https://www.watterott.com/de/ESP8266-ESP-07S>, Letzter Zugriff: 28.06.2019.
- [6] Briefkasten Git-Repository, <https://github.com/esdkrwl/UpcycledMailbox>, Letzter Zugriff: 28.06.2019.
- [7] Mosquitto, <https://mosquitto.org/>, Letzter Zugriff: 28.06.2019.
- [8] Node-RED, <https://nodered.org/>, Letzter Zugriff: 28.06.2019.
- [9] Telegram, <https://telegram.org/>, Letzter Zugriff: 28.06.2019.
- [10] Twitter, <https://twitter.com/>, Letzter Zugriff: 28.06.2019.
- [11] Upcycling, <https://en.wikipedia.org/wiki/Upcycling>, Letzter Zugriff: 28.06.2019.

Upcycling eines Briefkastens

Anatolij Fandrich

UPCYCLING?

- Briefkasten optisch veraltet
- Funktionsumfang erweitern
 - Benachrichtigung via App bei neuer Post
 - Unnötige Wege zum Briefkasten vermeiden
 - Peace of Mind: Wichtige Sendungen sofort abholen
- Wertsteigerung durch Elektronik im Briefkasten

FUNKTIONSWEISE

- Magnetschalter an Klappe am Einwurfschlitz
- Mikrocontroller im Briefkasten verbindet sich mit WLAN, sobald sich Klappe bewegt
- Daten über Internet an Server gesendet, verarbeitet und weitergeleitet an Smartphone
- Zusätzlich Warnung bei geringen Akkustand
- Energiesparmodus nach dem Versenden



SYSTEMÜBERSICHT

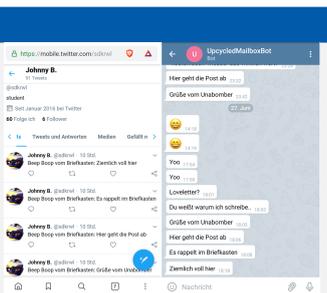
- Mikrocontroller befindet sich im Briefkasten
- Beim Einwurf von Post wird Akkustandmessung durchgeführt
- Verbindung zum WiFi-Router und zum Internet wird aufgebaut
- Akkustand wird via MQTT an Broker Mosquitto versendet
 - Broker mqtt.iot.informatik.unioldenburg.de
 - Topic `aamn/Briefkasten/1`
 - Payload `String[4]`
- Mosquitto leitet Daten an Node-RED weiter
- Via Node-RED kann mit Telegram, Twitter oder e-Mail eine Benachrichtigung versendet werden
- Nutzer erhält Benachrichtigung auf Smartphone

HARDWARE

- 32-Bit Mikrocontroller ESP8266EX im Modul ESP-07S
- Externe Antenne für Empfang innerhalb des Briefkastens
- LiFePo4-Akku, da dieser ohne Spannungsregler mit Mikrocontroller verbunden werden kann
- 10-Bit ADC und Spannungsteiler zur Akkuüberwachung
- Magnetschalter verbindet Mikrocontroller und Akku für Versand
- Nach Versand trennt Power Off Schaltung Mikrocontroller und Akku
- UART-Schnittstelle zum Programmieren und Debuggen
- Status-LED zum Debuggen
- ePaper-Schnittstelle für künftige Updates

ERGEBNISSE

- Einfacher Einbau in Briefkasten
- DIY Platine (siehe rechts) funktioniert
 - kompakt
 - für Akkubetrieb optimiert
 - sparsam (0 μ A im Standby)
- Benachrichtigung via App (siehe rechts)
- Vollständig online dokumentiert zum Nachmachen



GESAMTKOSTEN

Platine (unbestückt)	1,33€
Mikrocontroller ESP-07S	2,30€
Platine (voll bestückt)	7,60€
Antenne	5,95€
LiFePo4 Akku (1100mAh)	7,29€
Magnetschalter	2,36€
Briefkasten (optional)	19,90€
Summe	ca. 40€

AUSBLICK

- Die Montage eines ePaper-Displays mit Informationen für den Paketzusteller
- Entriegelung des Briefkastens mit Keypad und Pin oder Keyless mit Smartphone
- Platine mit THT statt SMD-Technologie bestücken, um Novizen den Nachbau zu vereinfachen
- Andere Benachrichtigungsmöglichkeiten erproben, wie z.B. Beleuchtungseinheiten
- Andere Funkstandards erproben, wie z.B. LoRa für ländliche Gebiete
- Kommunikation zwischen Broker und Briefkasten verschlüsseln
- Designfehler auf der Platine ausbessern (siehe Git-Repository unter Known-Issues)

WEITERE INFORMATIONEN

Die Ergebnisse dieses Projekts sind in einem Git-Repository abgelegt. Durch Scannen des QR-Codes erhält man u.a. Zugang zum Schaltplan, den Gerber-Files der Platine, der Firmware, den 3D-Modellen, weiteren Abbildungen, einer umfangreichen Dokumentation und einer Anleitung zur Inbetriebnahme.

Raspberry Pi VoIP Softphone

Sören Rempel
Fakultät 2
Systemsoftware und verteilte Systeme
Carl von Ossietzky Universität
Oldenburg

Abstract

Upcycling has been a part of recycling since the late 1990s. In this paper I describe a way to upcycle an old rotary-dial landline telephone with new components to make it compatible with digital telephony. A brief description of upcycling and the history of the telephone will be given. Afterwards I describe the soft- and hardware used for the project and discuss the difficulties and the result of the project.

1. Introduction

The term "Upcycling" was first described in the book "Upcycling" from Gunter Pauli in 1999/98 [1]. It stands for creative reuse of waste products. The new product normally should be of equal or higher value than the waste product. The general idea is to save the environment by using existing materials instead of using new ones.

Pauli is the founder of the ZERI. "Zero Emissions Research and Initiatives (ZERI) is a global network of creative minds, seeking solutions to the ever increasing problems of the world" [2]. ZERI is considered one of the biggest Think Tanks for innovative policies in the world [3].

My upcycling project is an old rotary-dial telephone that should be usable with new technologies.

2. Related Work

In general most rotary-dial telephones are upcycled in a creative way without still having the normal functions of a phone. People have made lamps [4] or clocks [5] out of old rotary-dial telephones.

For softphones with a Raspberry Pi there is quite a lot of related work. One of the most popular products realizing a phone with a Raspberry Pi is the ZeroPhone [6]. It is an open-source project that wants to spread cheap self-build

mobile phones to everyone. The ZeroPhone is based on a Raspberry Pi Zero costs roughly 50\$ in parts. The underlying architecture is Linux with an UI written in Python.

3. History of the telephone

The first patent for a telephone was granted to Alexander Graham Bell in 1876 [7]. It quickly became one of the most popular means of communication. In 1883 at the "Internationalen Elektrischen Ausstellung" [8] a new type of microphone and a better telephone receiver were presented, which marked a big milestone in the development of the telephone. The range and quality of phone calls was greatly improved. In 1892 a patent for the rotary-dial was granted [9] and made using telephones a lot easier. The technology was mostly used in telephones until the 1990s when a new technology, the dual-tone multi-frequency signaling [10] was adapted by more companies. Nowadays analog telephones are not used anymore and are mostly not compatible with new digital telephony.

4. Upcycling on the basis of a Krone FeTAp 792

For my project I choose an old analog telephone from the 1980s, the Krone FeTAp 792. It is a rotary-dial telephone used in the university in the 1980s and 1990s. The connection to the telephone network is not possible anymore with modern connectors. In the following chapters the process of the upcycling will be described.

5. Hardware Architecture

The main part of the upcycled telephone is a Raspberry Pi 3B. The described software in section 6 will be running on it and utilizing the GPIO Pins from the Pi. A lot of the necessary hardware can be reused from the FeTAp 792. The push button under the telephone receiver is used for accepting and ending calls and is connected to the Raspberry Pi



Figure 1. Krone FeTAp 792 [11]

via GPIO. The telephone receiver with the microphone and speaker is replaced by a USB VoIP telephone receiver from sipgate with an internal soundcard for better audio quality. For ringing, the old bell from the FeTAp 792 is used. The rotary-dial works with an Arduino Mini to register the dialed number. The button at the bottom of the telephone is used to turn the telephone on and off.

6. Software Architecture

On the Raspberry Pi the newest version of Raspian is installed. As a VoIP provider sipgate.de is used. This limits the softphones which can be used. For Linux there are 5 different softphones, that are frequently used [12]. The biggest limitation is that the telephone does not have a screen and only works with buttons or switches and therefore needs a console interface or a suitable interface. Most softphones are written for graphical use.

Ekiga

Ekiga is a softphone, which was part of the standard gnome-package for Ubuntu. It also provides support for video calls. The softphone only has a graphical interface and can not be used via the console.

Kphone

Kphone comes from the KDE package and provides a softphone without video calling or a console interface.

Linphone

Linphone is still supported version and provides a complex Python API for developing your own program. It provides all necessary functions, however the API does not work good with accounts that are not from Linphone directly and is quite complex, making it overloaded for the upcycling project.

Jami

Jami [13], formerly known as GNU Ring and SFLPhone, is one of the most advanced softphones and is still currently getting updates. It is compatible with Linux, Windows, iOS, macOS and Android. Jami provides SIP support, video calls, messaging and much more. It does not however provide a lightweight console interface that can be used for the upcycling project.

Twinkle

Twinkle [14] is the softphone chosen for the project. It is easy to setup with sipgate.de as the provider and has an easy to use command line interface that can be controlled via a Python script. For the project the newest version from 2016 is used.

7. Realization on the Raspberry Pi

On start-up a Python script is started that starts an instance of the Twinkle command line interface (CLI) and logs in the user with a previously configured account. The CLI can perform tasks such as call a number with `<call "number">`, send messages with `<message "user" "message">` and receive calls. When a call is received, the user can accept the call by lifting the telephone receiver. This triggers a button that automatically sends the answer command to the Twinkle CLI. The call can be ended by lowering the telephone receiver onto the telephone again and triggering the button again. Calling a number is done by dialing a number on the rotary-dial. The rotations of the dial are counted with a GPIO pin from the Raspberry Pi. When lifting the telephone receiver the call command with the number is written into the CLI.

8. Google Assistant

A new upcycled feature of the telephone is the integration into the Google Assistant world. The Raspberry Pi is a registered Google Assistant speaker and can be used to turn on the lights or check the weather. The Google Assistant was installed using the Google Python library for the Google Assistant and registered via the Google Cloud Platform. This enables the Google Assistant to communicate with any other devices, i. e. a smart lamp, also registered on your Google account.

9. Limitations

Choosing the Raspberry Pi as the basis for a softphone without a graphical interface limits the performance and usability. Finding a suitable softphone for the task, was the

hardest part. Although an external sound card is used, the quality of the call is not as high as with modern digital phones and may not even be as good as the old functioning version of the telephone. This also stands in relation with the choice of an Raspberry Pi, since many softphones do not support ARM versions. Twinkle mainly uses the GSM or G.711/G.726 codec which are still commonly used in VoIP telephones but have been replaced by better codecs. The codec from Twinkle works with low bandwidths and is the oldest audio codec. This codec ensures high audio qualities but is limited by the Raspberry Pi and the softphone used. The audio coming from the other party is received clearly and in high quality, however the outgoing audio is not received clearly by the other party and is delayed. This may be a problem of the softphone not being configured correctly for Raspbian / ARM architecture. The Google Assistant running purely with Python has good audio quality however while receiving and sending audio.

10. Conclusion

In conclusion there are a few things I could have done better. I would reconsider choosing the Raspberry Pi instead of another single-board computer. However a single-board computer without an ARM architecture comes with a significantly higher price. Choosing a different processor architecture may solve the problems with the softphones and the bad audio quality when sending signals. Sipgate is an easy to use and highly compatible VoIP provider and the audio quality provided by the sipgate receiver is good. The Google Assistant in contrast to the softphone was easy to setup and works perfectly.

References

- [1] Gunter Pauli, *Upcycling*, Riemann Verlag, 1999
- [2] zeri.org, *Zeri*, accessed 2019-06-22
- [3] <https://www.gunterpauli.com/zeri.html>, *Zeri*, accessed 2019-06-22
- [4] https://www.boredpanda.com/diy-rotary-phone-lamp-bjorn-freed/?utm_source=googleutm_medium=organicutm_campaign=organic, *Rotary Phone Lamp*, accessed 2019-06-22
- [5] <https://www.pinterest.de/pin/478437160413235179/>, *Phone Plant Pot*, accessed 2019-06-22
- [6] <https://zerophone.org/>, *ZeroPhone*, accessed 2019-06-22
- [7] https://worldwide.espacenet.com/publicationDetails/biblio?CC=USNR=174465KC=FT=Elocale=en_EP,US174465 Patent, 1876
- [8] <http://anno.onb.ac.at/cgi-content/anno?apm=0aid=nfpdatum=18831101seite=5,Neue Freie Presse>, 1883
- [9] <https://patents.google.com/patent/US486909,US486909 Patent>, accessed 2019-06-22
- [10] <http://www.elektronik-kompendium.de/sites/kom/0304292.htm>, *MFV*, accessed 2019-06-22
- [11] <https://www.wasser.de/telefon-alt/forum/forum-bilder/100000000006282-00-1244843903-g.jpg>, *Krone FeTAp 792*, accessed 2019-06-22
- [12] <https://www.linux-magazin.de/ausgaben/2009/08/quasselstrippen/>, *Different Softphones tested*, accessed 2019-06-27
- [13] <https://jami.net/>, *Jami Softphone*, accessed 2019-06-27
- [14] <https://wiki.ubuntuusers.de/Twinkle/>, *Twinkle Softphone*, accessed 2019-27-06

Upcycling anhand eines alten Telefons

Sören Rempel

UPCYCLING

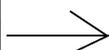
• Was versteht man unter Upcycling?

Beim Upcycling werden Abfallprodukte wiederverwertet, um aus diesen neue, "bessere" Gegenstände zu kreieren.

z.B. wird aus einem alten Telefon eine Lampe oder aus alten Blechdosen wird eine Skulptur



HARD- UND SOFTWARE REALISIERUNG



- Aufwerten eines alten Telefons, des Krone FeTap 792
- Keine künstlerische Aufwertung, sondern technische Aktualisierung
- Neues Herzstück des Telefons ist ein Raspberry Pi 3B
- Klingel, Wählscheibe und Taster vom Hörer werden weiter benutzt
- Austausch des alten Handapparats durch neuen USB-Handapparat von Sipgate
- Aus dem Festnetztelefon wird so ein Voice-over-IP Telefon mit Twinkle als Softphone
- Die Kommunikation zwischen der Hardware und dem Softphone erfolgt mithilfe eines Python Scripts über die Kommandozeile
- Als zusätzliches "modernes" Feature ist das Telefon gleichzeitig auch ein Google Assistant
- Die Realisierung erfolgt über die Google Assistant Python Library und kann in Google Home integriert werden

BENUTZERANLEITUNG

• Benutzung wie ein normales Telefon

- Dem Telefon ist die Nummer 0441/55979127 zugewiesen
- Annehmen eines Anrufs durch Abheben des Hörers
- Beenden des Telefonats durch Auflegen des Hörers
- Wählen einer Rufnummer mit der Wählscheibe

• Benutzung als Google Assistant

- Kommunikation erfolgt auf Englisch
- Gestartet wird diese mit dem Befehl "Hey Google" oder "OK Google"
- Anschließend können Fragen gestellt werden wie z.B. "How will the weather be tomorrow?"
- Der Google Assistant antwortet dann auf die gestellte Frage

KOSTEN

- Raspberry Pi 3B - ca. 32€
- Altes Wählscheibentelefon
 - Sammlerstücke - sehr schön, aber sehr teuer
 - Günstige Alternative: Altes Telefon von Oma oder aus dem Büro - 0€
- VoIP-Telefonhörer - ca. 10€
- Rufnummer von Sipgate - 0€
- Ausgehende Anrufe - 1,79ct pro Minute Festnetz, Mobil 14,90ct pro Minute

QUELLEN

- Upcycling-Logo: <https://banner2.kisspng.com/20180708/aju/kisspng-upcycling-recycling-symbol-plastic-bag-logo-sense-of-worth-5b42cc0e3dce54.1633497115311042702532.jpg>
- Sipgate.de-Logo: https://cdn2.downdetector.com/static/uploads/logo/sipgate_wort-bild-marke_RGB_schwarz.png
- Krone FeTap 792: <https://www.wasser.de/telefon-alt/forum/forum-bilder/10000000006282-00-1244843903-g.jpg>
- VoIP-Hörer: [https://i.ebayimg.com/00/s/NTY4WDEwMjQ=/z/jFsAAOS-wzNZbPicB/\\$.JPG](https://i.ebayimg.com/00/s/NTY4WDEwMjQ=/z/jFsAAOS-wzNZbPicB/$.JPG)
- Raspberry Pi 3B: https://cdn-reichelt.de/bilder/web/xxl_ws/A300/RASP_03_01.png

Spiegel mit Wetteranzeige durch LED-Hintergrundbeleuchtung

Sarah Katharina Thien

Proseminar "Aus alt mach neu: Upcycling mit Raspberry Pi und Co."

Carl von Ossietzky Universität
Oldenburg

Abstract

In diesem Projekt geht darum, einem Alltagsgegenstand durch Upcycling neue Funktionen zu geben. Ich habe mich für das Upcycling eines Spiegels entschieden. Mithilfe von LEDs wird der Spiegel zu einem smarten Spiegel, der dem Benutzer Informationen anhand von Licht darstellen kann. In diesem Fall handelt es sich um Wetterdaten. Diese werden bei Start des Spiegels aus einer Wetter-API abgefragt. Der Spiegel zeigt nacheinander erst das aktuelle Wetter in Form der Bewegung des Lichts und danach die Temperatur durch die Farbe des Lichts an. Auf Knopfdruck fragt der Spiegel erneut das aktuelle Wetter und die Temperatur ab und zeigt sie an.

1. Upcycling, was ist das überhaupt?

Upcycling ist eine Form des Recyclings. Recycling bedeutet, dass aus Abfall wieder neue gleichwertige Produkte entstehen. Recycling wird besonders für Rohstoffe wie Metall und Kunststoff eingesetzt. Durch die Wiederverwertung werden Rohstoffe gespart und die Umwelt geschont. [1]

Upcycling beschreibt ebenfalls eine Form der Wiederverwertung. Der Unterschied zum Recycling ist, dass der Wert der Gegenstände durch Upcycling erhöht wird. So werden aus alten Gegenständen höherwertige neue Gegenstände. Dabei haben die upgecycelten Gegenstände oft neue Funktionen und nur noch wenig mit dem ursprünglichen Gegenstand zu tun. [2] Im Folgenden werde ich meinen Upcycling-Gegenstand und Ideen zum Upcycling dieses Gegenstandes vorstellen. Dann gehe ich auf die einzelnen Phasen der Realisierung meines Projektes ein. Im Anschluss diskutiere ich das Ergebnis und gebe einen Ausblick.

2. Mein Upcycling-Gegenstand

Ich habe mich dafür entschieden, einen Spiegel upzucyclen. Da ein Spiegel wenige Funktionen hat, kann man

diesem Gegenstand durch den Einsatz eines Raspberry Pis oder ähnlichem eine Vielzahl an weiteren Funktionen geben. Ein Raspberry Pi ist ein Einplatinen-Minicomputer. Seine Größe entspricht circa der einer Kreditkarte. Er verfügt über Schnittstellen wie USB, Ethernet und GPIO-Pins, über die weitere Geräte mit dem Raspberry Pi verbunden werden können. GPIO-Pins sind General Purpose Input Output-Pins, an die z.B. Taster, Sensoren und LEDs angeschlossen werden können. Der Raspberry Pi Zero W, den ich in diesem Projekt verwenden werde, verfügt statt einer Ethernet-Schnittstelle über ein WLAN-Modul. [4] Im Folgenden werde ich den Spiegel als Upcyclinggegenstand vorstellen und mögliche neue Funktionen erläutern.

2.1. Entwicklungsgeschichte des Spiegels

Schon in den Mythen aus der griechischen Antike wird davon erzählt, dass Narziss sich selbst in der spiegelnden Wasseroberfläche betrachtete und von diesem Anblick fasziniert war. Diese Faszination der Notwendigkeit, sich selber sehen zu können, trieb die Entwicklung von Spiegeln wie wir sie kennen voran. Die ersten Spiegel bestanden aus poliertem Metall und wurden in Ägypten und Griechenland benutzt. Schon die Redner im alten Griechenland um ca. 400 v. Chr. übten ihre Reden vor dem Spiegel.

Der genaue Zeitpunkt der ersten Glasspiegel ist nicht bekannt, hergestellt wurden sie jedoch schon im 14. Jh. in Europa. Rund geblasenes Glas wurde auf der Rückseite mit Metall oder einer Metallfolie beschichtet. Flache Spiegel aus Kristallglas kamen dann in der Renaissance in Venedig auf und wurden in die ganze Welt exportiert.

Der Spiegel entwickelte sich vom kleinen Taschen- oder Handspiegel zum Einrichtungsgegenstand und wurde besonders in der Barockzeit sehr geschätzt. In den großen Schlössern wurden ganze Säle mit den damals sehr teuren Spiegeln ausgestattet. Ein berühmtes Beispiel dafür ist das Schloss Versailles von König Ludwig XIV. von Frankreich. Im 19. Jh. zog der Spiegel dann auch in die bürgerlichen Wohnungen ein. Heute sind Spiegel aus unserem Leben nicht mehr wegzudenken. [5]

2.2. Gebrauch des Spiegels

Ein Spiegel hat von sich aus nur wenige Funktionen. Die hauptsächliche Funktion ist es, ein Spiegelbild der Umgebung wiederzugeben. Spiegel werden in nahezu jedem Haushalt eingesetzt und sind aus dem täglichen Leben nicht mehr wegzudenken. Vom Zähneputzen morgens nach dem Aufstehen bis zum Blick in den Spiegel vor Verlassen des Hauses begleitet er uns den ganzen Tag über. Des weiteren begegnen uns Spiegel oft im Alltag, z.B. in Umkleidekabinen oder auch in Spiegelreflexkameras. In Autos sorgen sie für mehr Sicherheit, indem das Blickfeld des Fahrers durch die Spiegel erweitert wird. Dadurch werden andere Verkehrsteilnehmer rund um das Auto schneller erkannt.

2.3. Ideen zum Upcycling eines Spiegels

Ein Spiegel ist eine gute Grundlage für das kreative Upcycling, da er von sich aus nur wenige Funktionen bietet. Mithilfe von Raspberry Pi und Co. lässt sich so aus einem einfachen Spiegel ein smarter Spiegel bauen, der dem Nutzer die unterschiedlichsten Informationen anzeigen kann. Dies kann z.B. durch das Ersetzen der Spiegelfläche durch einen Bildschirm gelöst werden. Vor den Bildschirm kann eine lichtdurchlässige spiegelnde Plexiglasscheibe verbaut werden, sodass die Spiegelfunktion erhalten bleibt. [3]

Eine andere Art, einen smarten Spiegel umzusetzen, ist eine Hintergrundbeleuchtung. Dazu werden LED-Streifen auf die Rückseite des Spiegel geklebt, deren Licht dann an die Wand scheint und den Spiegel rundherum beleuchtet. So können mithilfe von unterschiedlichen Lichtfunktionen Informationen wie z.B. Wetterdaten angezeigt werden.

2.4. Entscheidung

Ich habe mich für die zweite Option entschieden, da ich gerne ein Projekt mit LED-Streifen realisieren wollte. Diese Variante hat auch den Vorteil, dass kein Bildschirm in den Rahmen des Spiegels eingesetzt werden muss. Der Spiegel bekommt durch die LEDs die neuen Funktionen, das aktuelle Wetter sowie die ungefähre aktuelle Temperatur darzustellen. Realisiert wird dies über Bewegung und Farbe des Lichts.

3. Realisierung

Im Folgenden erläutere ich die Realisierung meines Projekts "Spiegel mit Wetteranzeige durch LED-Hintergrundbeleuchtung".

3.1. Entwurf

Zuerst habe ich recherchiert, wie man LED-Streifen mit einem Raspberry Pi steuern kann und eine Liste aller be-

nötigten Teile erstellt. [8] Um ein passendes Schaltnetzteil zu finden, habe ich ausgerechnet, wie viele LEDs ich benötigen werde, um den Spiegel rundherum zu beleuchten. Eine LED benötigt 5V Spannung und bei voller Lichtintensität eine Stromstärke von 0,06A. [8] Ich verwende LED-Streifen mit 60 LEDs pro Meter. Bei einer Abmessung von 45cm * 45cm des Spiegels benötige ich einen LED-Streifen der Länge 180 cm, um den Spiegel von der Rückseite rundherum mit LEDs zu beleuchten. Daraus lässt sich berechnen, dass dieser Streifen $1,8\text{m} * 60 \text{ LEDs/m} = 108 \text{ LEDs}$ hat, woraus sich eine benötigte Gesamtstromstärke von $108 * 0,06\text{A} = 6,48\text{A}$ ergibt. Damit das Schaltnetzteil nicht seine Belastungsgrenze erreicht, habe ich ein Netzteil mit 5V und 10A verwendet.

3.2. Hardware

Im ersten Schritt habe ich WS2812B-LED-Streifen, auch NeoPixel genannt, auf die Rückseite eines Spiegels geklebt. Hierbei handelt es sich um RGB-LED-Streifen, dessen LEDs eine Kontrolllogik in Form des WS2811-Controllers enthält. Dadurch sind die LEDs einzeln adressierbar. Jeder Pixel nimmt sich aus den Daten nur die für ihn relevanten Informationen und schickt den Rest an die nachfolgenden Pixel weiter. Die an die WS2811-Controller gesendeten Datenpakete haben eine Größe von 24 Bit, je acht Bit für die Farben Rot, Grün und Blau. In einem WS2812B-LED-Streifen fließen somit Sequenzen von 24-Bit-Paketen über die Datenleitung. [9]

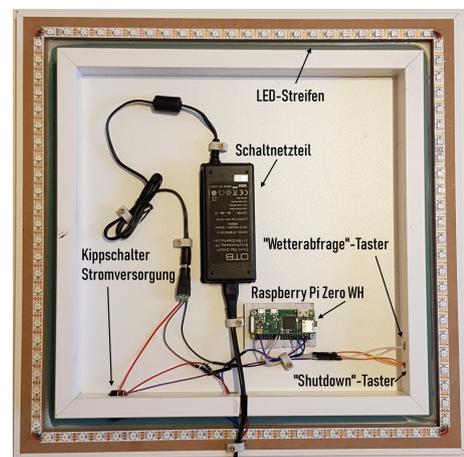


Abbildung 1. Aufbau des Spiegels

Die LED-Streifen habe ich auf die passende Länge geschnitten und möglichst nah an den Rand des Spiegels geklebt. An den Ecken habe ich die einzelnen Streifen verlötet, sodass sie sich wie ein langer Streifen verhalten. Bei einem Funktionstest der LEDs habe ich festgestellt, dass

es sich GRB- statt um RGB-LEDs handelt. Das heißt, dass die Farben in der Reihenfolge Grün-Rot-Blau an die LEDs übergeben werden müssen.

Über ein Schaltnetzteil, welches über einen Steckeradapter mit den Spannungskabeln des LED-Streifens verbunden ist, werden die LEDs mit 5V Spannung versorgt. Ebenso wird der Raspberry Pi über dieses Schaltnetzteil mit Spannung versorgt. [8] Den Raspberry Pi habe ich über Jumper-Kabel mit VCC und Ground des Schaltnetzteils verbunden. Die 5V Versorgungsspannung bekommt der Raspberry Pi über Pin 1, Masse wird an Pin 3 angeschlossen. Der Datenausgang für die LEDs ist Pin 6 (BCM Pin 18), der mit der Datenleitung des LED-Streifens verbunden wird. Die Masse der LEDs ist an einen Masse-Pin des Raspberry Pis angeschlossen. Die Daten für die LEDs werden über BCM Pin 18 an die LED-Streifen übergeben. Dieser Pin eignet sich besonders gut für die Datenübertragung an LED-Streifen, da hierüber sehr genaue zeitliche Signale möglich sind.¹

Gesteuert werden die LEDs über einen Raspberry Pi Zero WH, auf dem ein Python-Skript ausgeführt wird, das die Wetterdaten für gegebene Koordinaten aus einer Wetter-API abfragt.

Um den Raspberry Pi hinter dem Spiegel zu verstecken, habe ich, wie in Abbildung 1 zu sehen, einen Holzrahmen auf die Rückseite des Spiegels geklebt. Darin habe ich auch das Schaltnetzteil festgeklebt. An der Seite des Rahmens habe ich einen Kippschalter eingebaut, der die Stromversorgung zwischen Schaltnetzteil und Raspberry Pi bzw. LED-Streifen unterbricht.

Zusätzlich habe ich noch zwei Taster an die Seite des Spiegels eingebaut.² Der Raspberry Pi besitzt keinen Knopf, mit dem er heruntergefahren werden kann. Aus dem Grund habe ich einen der Taster dafür vorgesehen, den Raspberry Pi auf Knopfdruck herunterfahren zu können. Dies wird über ein Python-Skript realisiert, das im Hintergrund ausgeführt wird. Der andere Taster hat die Funktion, das aktuelle Wetter abzufragen und anzuzeigen.

Die Taster werden an die Pins BCM 11 bzw. BCM 21 und Ground angeschlossen. GPIO-Eingänge sind in der Regel undefiniert und springen zwischen “Low” und “High” hin und her. Daher verseehe ich die Eingangspins über das Python-Skript mit einem Pull-Up-Widerstand. Dadurch wird der GPIO-Eingang bei offenem Schalter gegen +VCC gezogen, sodass “High” (+3,3V) anliegt. Wird der Schalter geschlossen, liegt Ground und somit “Low” an. [6] Sobald an einem der Taster “Low” anliegt, wird ein Skript ausgeführt, das durch das Signal des Tasters aufgerufen wird.

¹https://de.pinout.xyz/pinout/pin12_gpio18

²<https://www.quartoknows.com/page/raspberry-pi-shutdown-button>

3.3. Software

Zuerst habe ich ein Image des Betriebssystems Raspbian Stretch³ via Balena Etcher⁴ auf eine Micro-SD-Karte geflasht. Dies habe ich anhand einer Anleitung [10] realisiert. Um über ssh auf den Raspberry Pi zugreifen zu können, habe ich als nächstes eine *wpa_supplicant.conf*-Datei im Ordner boot auf der SD-Karte angelegt und eine leere Datei mit dem Namen *ssh* hinzugefügt, um die erste ssh-Verbindung mit dem Raspberry Pi zu ermöglichen. In die *wpa_supplicant.conf*-Datei habe ich die Zugangsdaten für das WLAN geschrieben [10]. Das Ausführen des Programms hat gezeigt, dass die LEDs des Streifens alle funktionieren, jedoch nicht die Standard-Konfiguration RGB, sondern GRB haben.

Der nächste Schritt war das Anlegen eines Accounts bei der DarkSky-API.⁵ API bedeutet “Application Programming Interface” und ist eine Programmierschnittstelle, die auf der Quelltext-Ebene angebunden wird. [7] Die DarkSky-API ermöglicht 1000 kostenlose Wetterabfragen pro Tag. Auf der Website gibt es eine Anleitung zum Abfragen der Wetterdaten.⁶ Damit habe ich ein kleines Programm geschrieben, siehe Abbildung 2, um die Abfrage der Wetterdaten zu testen. Die Wetterdaten werden in einem JSON-String gespeichert. Aus diesem können die gewünschten Daten ausgelesen werden.

```
#!/usr/bin/python3

# http://www.electronics-lab.com/
# temperature-controlled-stair-lights-with-raspberry-pi
}# https://darksky.net/dev/docs

}from urllib.request import urlopen
import json
}import time

# get a key from https://developer.forecast.io/register
apikey = "b40c04a22b2c0dd5ed5f5af9242ad7a5"
# latitude & longitude University of Oldenburg
lati = "53.146565"
longi = "8.181729"

# get the data from the api website
url = "https://api.darksky.net/forecast/" + apikey\
      + "/" + lati + "," + longi + "?units=auto"
meteo = urlopen(url).read()
meteo = meteo.decode('utf-8')
weather = json.loads(meteo)

# get current temperature
currentTemp = weather['currently']['temperature']
# get current weather
currentWeather = weather['currently']['icon']
```

Abbildung 2. Code-Ausschnitt: Abfrage der Wetter-API

³<https://www.raspberrypi.org/downloads/raspbian/>

⁴<https://www.balena.io/etcher/>

⁵<https://darksky.net/dev>

⁶<https://darksky.net/dev/docs>

Dieser sieht wie folgt aus: 'latitude': 53.146565, 'longitude': 8.181729, 'timezone': 'Europe/Berlin', 'currently': 'time': 1560718053, 'summary': 'Partly Cloudy', 'icon': 'partly-cloudy-night', 'precipIntensity': 0, 'precipProbability': 0, 'temperature': 15.76, 'apparentTemperature': 15.77, 'dewPoint': 14.31, 'humidity': 0.91, 'pressure': 1021.28, 'windSpeed': 1.78, 'windGust': 1.78, 'windBearing': 74, 'cloudCover': 0.14, 'uvIndex': 0, 'visibility': 10.041, 'ozone': 329.1, ...

Das Programm, das den LED-Streifen anhand der Wetterdaten steuern soll, habe ich in Python 3 geschrieben. Von der Wetter-API kann ich das aktuelle Wetter abfragen, das mir Werte wie z.B. "cloudy" oder "rainy" zurückgibt. Diese stelle ich dann anhand von Funktionen, die aufgerufen werden, mithilfe der LEDs dar. Dafür habe ich mir sowohl eigene Funktionen geschrieben als auch Funktionen aus dem Beispielpogramm `strandtest.py` der `rpi_ws281x`-Library⁷ umgeschrieben. Ebenso frage ich die aktuelle Temperatur ab, die, wie in Abbildung 3.3 zu sehen, durch Farben dargestellt wird. Von -10 Grad in Dunkelblau geht es in Fünferschritten hoch über Grün, Gelb und Orange bis zu +39 Grad in Rot. Für den Fall, dass die aktuelle Temperatur kälter als -10 Grad oder wärmer als +39 Grad ist, habe ich die Default-Farbe Pink vorgesehen. Orientiert habe ich mich in der Farbgebung an einem Projekt, das mich auf die Idee dieses Upcycling-Projekts gebracht hat. [11]

Temperatur	Farbe
-10 °C bis -6 °C	Dunkelblau
-5 °C bis -1 °C	Dunkelblau
0 °C bis 4 °C	Hellblau
5 °C bis 9 °C	Grün
10 °C bis 14 °C	Hellgrün
15 °C bis 19 °C	Grün
20 °C bis 24 °C	Gelb
25 °C bis 29 °C	Orange
30 °C bis 34 °C	Rot
35 °C bis 40 °C	Rot
default	Pink

Abbildung 3. Darstellung der Temperatur durch Farben

⁷<https://github.com/rpi-ws281x/rpi-ws281x-python.git>

Der Ablauf des Programms sieht es vor, dass zuerst kurz das aktuelle Wetter durch die Bewegung und Farbe des Lichts dargestellt wird und kurz danach die aktuelle Temperatur durch die zugehörige Farbe. In Abbildung 3.3 ist aufgelistet, welche Wetterphänomene dargestellt werden und wie die Darstellung realisiert wird.

Wetter	Darstellung des Wetters durch Licht
Klarer Tag	Hellblau mit gelben Sonnenstrahlen
Klare Nacht	Dunkelblau mit aufblinkenden „Sternen“
Regen	Blaue Streifen, die am Rand des Spiegels herunterlaufen
Schnee	Weißer Schneeflocken, die am Rand des Spiegels herunterfallen
Wind	Helles Licht, das sich um den Spiegel herumdreht
Bewölkt/ Nebel	Gräulich weißes Licht, das sich wie Wolken um den Spiegel herumbewegt
Teilweise bewölkt	Blaues und gräulich weißes Licht, das sich wie Wolken um den Spiegel herumbewegt

Abbildung 4. Darstellung des Wetters

Durch Einschalten des Spiegels und damit des Raspberry Pis wird das Programm über einen cron-Auftrag ausgeführt. Durch `@reboot` wird das Programm bei jedem Start des Raspberry Pis ausgeführt. Die erneute Abfrage des Wetters erfolgt auf Knopfdruck. Sobald der Pin signalisiert, dass der Taster gedrückt wurde, wird erneut das Wetter aus der Wetter-API abgefragt und somit immer das aktuelle Wetter angezeigt. Nach Anzeige des Wetters und der Temperatur bleibt die Farbe der Temperatur bestehen bis das Wetter erneut abgefragt oder die Stromversorgung unterbrochen wird. Das heißt, dass die Daten auch nach Herunterfahren des Raspberry Pis weiterhin an den LEDs anliegen.

4. Test und Diskussion

Der Spiegel zeigt die Wetterfunktionen und Temperaturen korrekt an. Viele der Wetterfunktionen sind deutlich zu erkennen. Einige der Funktionen können noch verfeinert werden, um das Wetter präziser und deutlicher anzuzeigen, da nicht jede Wetterfunktion auf den ersten Blick verständlich ist. Der Spiegel hat zwei Taster an der Seite, mit denen der hinter dem Spiegel versteckte Raspberry Pi heruntergefahren bzw. das aktuelle Wetter erneut angezeigt werden kann. Der Shutdown-Button erfüllt seine Funktion zuverlässig und fährt den Raspberry Pi herunter. Die Daten liegen

danach weiterhin an den LEDs an, sodass durch das Herunterfahren des Raspberry Pis im richtigen Moment auch eine bestimmte Lichtfunktion "eingefroren" werden kann. Bei dem Taster, der das aktuelle Wetter erneut anzeigen soll, gibt es teilweise unsaubere Daten, die an die LEDs gehen, sodass diese mehrere Funktionen gleichzeitig anzeigen wollen und dadurch undefiniert flackern.

5. Zusammenfassung und Ausblick

Der Spiegel hat durch das Upcycling zwei neue Funktionen gewonnen. Er kann das aktuelle Wetter und die ungefähre aktuelle Temperatur über LEDs auf der Rückseite visuell darstellen. Gesteuert werden die LEDs über ein Python-Skript, das auf dem Raspberry Pi ausgeführt wird. Das Skript fragt bei jeder Neuausführung die aktuellen Wetterdaten aus einer Wetter-API ab und führt die entsprechenden Funktionen zur Visualisierung aus. An der Seite des Spiegels befinden sich zwei Taster. Bei Betätigung des oberen Tasters wird erneut das aktuelle Wetter abgefragt und dargestellt. Der untere Taster fungiert als Shutdown-Button. Nach Betätigung des Shutdown-Buttons wird der Raspberry Pi heruntergefahren.

Die Abfrage der Wetter-API liefert deutlich mehr Werte als momentan mit dem Spiegel dargestellt werden. Es lässt sich nicht nur das aktuelle Wetter, sondern auch eine Vorschau für die ganze Woche abfragen. Mit einem weiteren Taster an der Seite des Spiegels könnte das Programm z.B. auch den Wetterbericht für den nächsten Tag oder die ganze nächste Woche anzeigen.

Insgesamt hat der Spiegel durch das Upcycling an nützlichen Funktionen dazugewonnen. Durch die Abfrage der Wetter-API bietet der Spiegel allerdings noch eine Vielzahl an weiteren Informationen, die über die LEDs dargestellt werden könnten.

Literatur

- [1] E. Feess, E. Günther, "Recycling", *Gabler Wirtschaftslexikon* Februar 2018, <https://wirtschaftslexikon.gabler.de/definition/recycling-44989/version-268290> (aufgerufen am 25.05.2019).
- [2] A. Martin, "Was ist Upcycling? Einfach erklärt" *Focus Online*, September 2015, https://praxistipps.focus.de/was-ist-upcycling-einfach-erklart_45820 (aufgerufen am 25.05.2019).
- [3] T. Danker, "Smart Mirror selbst bauen" *Glan-cr*, <https://glancr.de/smart-mirror-selbst-bauen/> (aufgerufen am 25.06.2019).
- [4] S. Luber, S. Augsten, "Was ist der Raspberry Pi?" *DevInsider*, Mai 2017, <https://www.dev-insider.de/was-ist-der-raspberry-pi-a-598750/> (aufgerufen am 16.06.2019).
- [5] B. Vaupel, "Eine kleine Kulturgeschichte des Spiegels, Der Zauber des Widerscheins", *Monumente Magazin für Denkmalkultur Deutschland Online*, April 2009, <https://www.monumente-online.de/de/ausgaben/2009/2/der-zauber-des-widerscheins.php#.XOMd2sgzbD4> (aufgerufen am 21.05.2019).
- [6] Autor, "Raspberry Pi: GPIO mit Pullup- oder Pulldown-Widerstand beschalten?", *Elektronik Kompendium*, Mai 2019, <https://www.elektronik-kompendium.de/sites/raspberry-pi/2006051.htm> (aufgerufen am 17.06.2019).
- [7] S. Luber, S. Augsten, "Definition „Programmierschnittstelle“ Was ist eine API?", *DevInsider*, März 2017, <https://www.dev-insider.de/was-ist-eine-api-a-583923/> (aufgerufen am 17.06.2019).
- [8] F. Stern, "WS2812 RGB LED Streifen per Raspberry Pi steuern", *Raspberry Pi Tutorials*, <https://tutorials-raspberrypi.de/raspberry-pi-ws2812-ws2811b-rgb-led-streifen-steuern/> (aufgerufen am 26.05.2019).
- [9] M. Stal, "Von Erleuchtungen und Lichterketten", *heise Developer*, August 2016, <https://www.heise.de/developer/artikel/Von-Erleuchtungen-und-Lichterketten-3277261.html> (aufgerufen am 21.06.2019).
- [10] T. Moody (Macgyver), "How to Setup Raspberry Pi Zero WH", *YouTube*, Februar 2019, <https://www.youtube.com/watch?v=3VO4vGIQ1pg> (aufgerufen am 26.05.2019).
- [11] Rik, "Temperature Controlled Stair Lights with Raspberry Pi", *electronics-lab*, Juli 2017, <http://www.electronics-lab.com/temperature-controlled-stair-lights-with-raspberry-pi/> (aufgerufen am 26.05.2019).

Spiegel mit Wetteranzeige durch LED-Hintergrundbeleuchtung

Sarah Katharina Thien

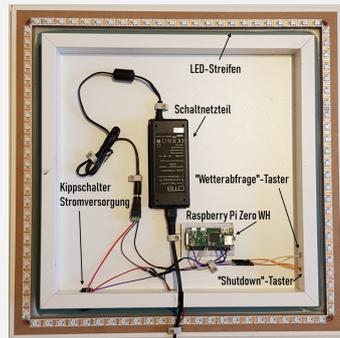
UPCYCLINGIDEEN FÜR EINEN SPIEGEL

- Smart Mirror mit Bildschirm
 - Einbau eines Bildschirms in den Rahmen
 - spiegelnde Plexiglasscheibe vor dem Bildschirm, um die Spiegelfunktion zu erhalten
 - Darstellung von Informationen auf dem Bildschirm
 - z.B. Nachrichten
 - Smart Mirror durch Hintergrundbeleuchtung mit LEDs
 - LED-Streifen hinter dem Spiegel
 - Licht der LEDs scheint an die Wand
 - Darstellung von Information über Farbe und Bewegung des Lichts
 - z.B. Wetterdaten
- ⇒ für diese Lösung habe ich mich entschieden



REALISIERUNG

- Hardware
 - WS2812B-LED-Streifen auf Rückseite des Spiegels
 - Raspberry Pi Zero WH führt Python-Skript aus
 - Stromversorgung von Raspberry Pi und LEDs über Schaltnetzteil
 - ein Taster für die Abfrage des aktuellen Wetters
 - ein Taster für das Herunterfahren des Raspberry Pis



- Software
 - Python-Skript(Pseudo-Code)


```
lade String mit Wetterdaten aus der API
currentWeather = Wetter['currently']['icon']
currentTemp = ...
if currentWeather == ... then führe
zugehörige Wetterfunktion aus
if currentTemperature == ... then führe
zugehörige Temperaturfunktion aus
if Wetterabfrage-Button pressed then lade
aktuelles Wetter aus der API und zeige es an
```

FUNKTIONEN DES SPIEGELS

- Anzeige des aktuellen Wetters über die Bewegung des Lichts
- Anzeige der ungefähren aktuellen Temperatur durch die Farbe des Lichts
- Anzeige von Wetter und Temperatur nacheinander
- Halten der Farbe der Temperatur bis zur nächsten Wetterabfrage
- erneute Abfrage und Anzeige des aktuellen Wetters und der Temperatur auf Knopfdruck
- Herunterfahren des Spiegels auf Knopfdruck
- "Einfrieren" der LEDs mit dem Herunterfahren-Knopf

• Anzeige des Wetters

Wetter	Darstellung des Wetters durch Licht
Klarer Tag	Hellblau mit gelben Sonnenstrahlen
Klare Nacht	Dunkelblau mit aufblinkenden „Sternen“
Regen	Blaue Streifen, die am Rand des Spiegels herunterlaufen
Schnee	Weißer Schneeflocken, die am Rand des Spiegels herunterfallen
Wind	Helles Licht, das sich um den Spiegel herumdrehet
Bewölkt/ Nebel	Grünlich weißes Licht, das sich wie Wolken um den Spiegel herumbewegt
Teilweise bewölkt	Blaues und grünlich weißes Licht, das sich wie Wolken um den Spiegel herumbewegt

• Anzeige der Temperatur

Temperatur	Farbe
-10 °C bis -6 °C	Blau
-5 °C bis -1 °C	Dunkelblau
0 °C bis 4 °C	Hellblau
5 °C bis 9 °C	Grün
10 °C bis 14 °C	Hellgrün
15 °C bis 19 °C	Grünlichgelb
20 °C bis 24 °C	Gelb
25 °C bis 29 °C	Orange
30 °C bis 34 °C	Rot
35 °C bis 40 °C	Rotrot
default	Magenta

BEISPIELE FÜR ANWENDUNGSFÄLLE

• klare Nacht



• 15 °C - 19 °C



• teilweise bewölkt



• 30 °C - 34 °C



VERWENDETE QUELLEN

- <https://glancr.de/smart-mirror-selbst-bauen/>
- <http://www.electronics-lab.com/temperature-controlled-stair-lights-with-raspberry-pi/>
- <https://tutorials-raspberrypi.de/raspberry-pi-ws2812-ws2811b-rgb-led-streifen-steuern/>
- <https://www.raspberrypi.org/downloads/raspbian/>
- <https://www.balena.io/etcher/>
- <https://darksy.net/dev>
- <https://darksy.net/dev/docs>
- <https://www.youtube.com/watch?v=3VO4vGIQ1pg>
- <https://github.com/rpi-ws281x/rpi-ws281x-python.git>
- <https://www.quartoknows.com/page/raspberry-pi-shutdown-button>
- <https://www.raspberrypi.org/documentation/linux/usage/cron.md>

Aus alt mach neu - Aufwerten einer Thermoskanne mithilfe eines Wemos D1 mini

Jana Gerdes
Department für Informatik
Carl von Ossietzky Universität Oldenburg

Abstract

Im Rahmen des Seminars "Aus alt mach neu: Upcycling mit Raspberry Pi und Co." ist die Idee entstanden, eine Thermoskanne mithilfe eines Wemos D1 mini sowohl optisch als auch funktional aufzubessern. In diesem Paper wird der Prozess dieser Aufbesserung dargestellt und erläutert.

1. Einleitung

In einer immer umweltbewusster werdenden Gesellschaft hört man immer wieder das Wort "Recycling" und kann sich darunter etwas vorstellen. Auch sieht man häufig das Recycling Symbol, das zeigt, dass bereits benutzte Stoffe genutzt werden, um aus ihnen etwas Neues aus demselben Material herzustellen.

Was aber bedeutet in diesem Zusammenhang "Upcycling"? Das Upcycling ist eine Art Erweiterung von Recycling. Auch hier werden bereits benutzte Stoffe verwendet, jedoch wird aus ihnen nichts Neues aus demselben Material oder mit derselben Funktionalität hergestellt. Die Haupteigenschaft des Upcyclings ist das Aufwerten des betreffenden Materials. Diese Aufwertung kann dabei funktional, qualitativ oder ökologisch sein. (vgl. [1])

In diesem Paper wird es um die Aufwertung einer Thermoskanne gehen. Da diese in fast jedem Haushalt zu finden ist, stellt eine Aufwertung dieses Gegenstandes für viele Menschen eine Bereicherung dar.

2. Klassischer Gebrauch einer Thermoskanne

Eine Thermoskanne ist "ein Behälter zum Warm- oder Kühlhalten von Speisen und Getränken" (siehe [2]). Sie wird auch "Isolierkanne" oder "Thermosflasche" genannt. "Thermobecher" sind kleinere Thermoskannen, die häufig mit einem Deckel, aus dem man direkt trinken kann, ausgestattet sind. Des Weiteren gibt es noch "Thermobehälter",

die für Speisen und Suppen gedacht sind. Egal in welcher Variante, Thermoskannen sollen die Temperatur ihres Inhaltes konstant halten, sodass dieser transportiert werden kann ohne abzukühlen bzw. warm zu werden. Am Häufigsten ist dieser Inhalt Tee oder Kaffee, man könnte jedoch auch Kakao oder Suppe in einer Thermoskanne transportieren oder im Sommer Wasser in ihr kühl halten.

3. Entwicklungsgeschichte

Entwickelt wurde die Thermoskanne im 19. Jahrhundert. Jedoch gab es nicht nur einen Erfinder, sondern gleich drei Männer, die die Idee hatten, die Temperatur von Flüssigkeiten konstant halten zu wollen (vgl.[3]). "1904 ließ [Reinhold] Burger den Namen "Thermosflasche" als Warenzeichen schützen" (siehe [4]) und ist daher oft als der Erfinder der Thermoskannen in verschiedener Literatur zu finden. Vor ihm hatten jedoch auch "der Chemnitzer Physikprofessor Adolf Ferdinand Weinhold bereits in den 1870er Jahren ein derartiges Isoliergefäß erdacht" (siehe [3]). Nach James Dewar, wurde "das Dewar-Gefäß" benannt und "die [...] Thermosflasche stellt ein solches Dewar-Gefäß dar" (siehe [5]), somit ist auch er einer der Erfinder der Thermoskanne.

Es gibt verschiedene Bauweisen von Thermoskannen, die jedoch alle eins gemeinsam haben. Das Konstanthalten der Temperatur in der Thermoskanne wird durch ein "doppelwandige[s], [...] Gefäß" (siehe [2]) ermöglicht, dessen Wände "durch eine Vakuumschicht voneinander getrennt sind" (siehe [3]). Dabei können diese doppelwandigen Gefäße aus Glas oder Edelstahl sein, wobei letzteres Material unempfindlicher gegenüber Stöße ist. Aus diesem Grund besteht die hier beschriebene Thermoskanne, die aufgewertet wurde, ebenfalls aus Edelstahl. Bei Thermoskannen aus Glas gibt es außerdem die Variante, bei der das Innengefäß verspiegelt ist, was "die Wärmeübertragung durch Wärmestrahlung" (siehe [5]) verhindert.

4. Upcycling Ideen

Am Häufigsten wird eine Thermoskanne heutzutage, wie eben beschrieben, benutzt, um mit ihr Tee oder Kaffee warmzuhalten und aus ihr trinken zu können. Beide Flüssigkeiten müssen dabei vorab zubereitet werden, wofür Wasser gekocht werden muss. Wenn man also unterwegs ist und keinen Wasserkocher zur Verfügung hat, kann die Thermoskanne auch nicht für Kaffee oder Tee genutzt werden. Aus diesem Grund könnte die Thermoskanne aufgebessert werden, indem man die Möglichkeit implementiert, die Temperatur der Flüssigkeit in ihr zu steuern. So könnte die Thermoskanne die Temperatur der Flüssigkeit nicht nur konstant halten, sondern diese bei Bedarf auch wieder zum Kochen bringen oder abkühlen.

Des Weiteren könnte die Möglichkeit implementiert werden, eine Zeit einzuprogrammieren, zu der der Inhalt die bestimmte Temperatur erreichen soll. So könnte die Thermoskanne den morgendlichen Kaffee schon fertig zubereitet haben, bevor man überhaupt aufgestanden ist und die in ihr mitgebrachte Suppe pünktlich zum Mittag auf die zum Verzehren perfekte Temperatur erhitzen.

Diese Idee könnte auch noch weiter ausgebaut werden, indem man aus der Thermoskanne eine Art Kochhelfer baut, der mithilfe einer Anzeige eine Anleitung gibt, welche Zutaten beim Kochen einer Suppe beispielsweise wann hinzugegeben werden müssen. So könnte man in der Thermoskanne nicht nur Flüssigkeiten transportieren, sondern auch Speisen in ihr nach Anleitung zubereiten.

Eine weitere Möglichkeit der Aufbesserung könnte das Implementieren einer Füllstandanzeige sein, die von außen sichtbar machen lässt, wie viel Inhalt noch in der Thermoskanne vorhanden ist. Das ist nützlich, da Thermoskannen in der Regel blickdicht sind, sodass der Füllstand nur durch Aufschrauben und mit einem Blick in das Gefäß ermittelt werden kann.

Die Füllstandanzeige könnte schließlich noch mit einer Anzeige der Temperatur und der Art der Flüssigkeit in der Thermoskanne verbunden werden. Mit dieser Art Aufbesserung kann verhindert werden, dass man sich beim Trinken aus der Thermoskanne an einem zu heißen Getränk den Mund verbrennt oder einen Tee oder Kaffee trinkt, der schon kalt geworden ist und daher nicht mehr so gut schmeckt. Außerdem könnte so beim Kaffeetrinken auf Geburtstagspartys beispielsweise, auf denen es Kaffee und Tee in Thermoskannen gibt, verhindert werden, dass jemand versehentlich Tee eingesehen bekommt, obwohl er lieber Kaffee getrunken hätte, weil die Thermoskannen verwechselt wurden.

Abschließend könnte man die Thermoskanne als Aufwertung smart gestalten, sodass die Informationen wie Füll-

stand oder Temperatur nicht nur an der Thermoskanne sichtbar werden, sondern über das Smartphone abrufbar sind.

4.1. Beschreibung der ausgewählten Idee

Im Rahmen dieses Seminars soll eine Thermoskanne aufgebessert werden, indem eine Möglichkeit geschaffen wird, die Temperatur der in der Thermoskanne befindlichen Flüssigkeit von außen sichtbar darzustellen. Dies soll zum Einen durch die Anzeige auf einem Display, der unter die Thermoskanne gebaut wird, geschehen. Zum Anderen wird ein LED Streifen von außen an die Thermoskanne geklebt und anhand der Farbe der LED kann dann ebenfalls die Temperatur abgelesen werden: der LED Streifen leuchtet rot, wenn die Flüssigkeit noch zu heiß zum Trinken ist, er leuchtet grün, wenn die Flüssigkeit die perfekte warme Temperatur hat, er leuchtet gelb, wenn die Flüssigkeit noch lauwarm ist und leuchtet blau, wenn die Flüssigkeit bereits zum Genießen zu kalt ist.

5. Realisierung und Implementierung

5.1. Hardwarearchitektur

Die Entscheidung bezüglich der Hardware, genauer für den Mikrocontroller, auf dem die Software für die Realisierung des Vorhabens laufen soll, ist auf einen Wemos D1 mini gefallen (siehe Nummer 7 von Abbildung 2). Dieser hat den Vorteil, dass er eine geringe Baugröße hat und außerdem ein WIFI Modul mitliefert, sodass die Möglichkeit besteht, die Realisierung über den Rahmen, in dem sie in diesem Seminar ausgearbeitet wurde, noch weiter auszubauen und die Thermoskanne W-Lan fähig zu machen. So könnte wie zuvor beschrieben eine weitere Aufwertung der Thermoskanne darin bestehen, dass mit dem Smartphone über eine App die Temperatur der in der Thermoskanne befindlichen Flüssigkeit abgerufen werden kann.



Abbildung 1. Bauteilliste

Die größte Herausforderung in diesem Projekt war das Finden einer für die Realisierung der Idee passenden Thermoskanne. Zuerst war die Idee, die Temperatur an dem Behälter der Thermoskanne von außen zu messen, sodass eine Thermoskanne nötig war, die aus zwei Behältern zusammengesetzt ist. Diese Idee wurde jedoch mit der Zeit verworfen. Anstelle dessen wird die Temperatur nun mit einem Temperaturfühler gemessen, genauer mit einem DS18B20, der wasserdicht ist und mithilfe eines Strohhalmes in die Flüssigkeit gelangt (siehe Abbildung 3). Der Vorteil an der nun für das Projekt ausgewählten Thermoskanne besteht darin, dass diese mit zwei Verschlussdeckeln geliefert wurde, von denen ein Verschlussdeckel schon mit einem Strohhalm ausgestattet war (siehe Nummer 1 und 12 von Abbildung 2).



Abbildung 2. Der verbaute Temperaturfühler

Um das für das Projekt ausgewählte Display (siehe Nummer 6 von Abbildung 2), auf dem die Temperatur der in der Thermoskanne befindlichen Flüssigkeit angezeigt werden soll, unter der Thermosflasche zu verbauen, wird ein Halter benötigt, der das Verbauen ermöglicht und in dem außerdem die Powerbank (siehe Nummer 5 von Abbildung 2), die den Mikrocontroller über ein USB-Kabel (siehe Nummer 9 von Abbildung 2) mit Energie versorgen soll, passt. Zuerst war die Idee, eine alte Orangensaft Flasche aufzuschneiden und diese über die Thermoskanne zu stülpen. Um den Halter jedoch optisch noch etwas hochwertiger aussehen zu lassen, wurde er nun in einem CAD Programm mit allen benötigten Aussparungen für die Verkabelung modelliert und an einem 3D-Drucker (Ultimaker 2+) gedruckt (siehe Abbildung 4).

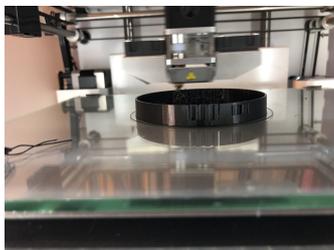


Abbildung 3. Der Halter wird gedruckt

5.2. Softwarearchitektur

Der Wemos D1 mini wurde in der Arduino Umgebung programmiert. Dafür wurden für die LED die Bibliothek "FastLED", für den Temperaturfühler die Bibliotheken "DallasTemperature" und "OneWire" und für das Display "Adafruit_SSD1306" genutzt. In dem Programm, das auf den Mikrocontroller gespielt wird, werden zuerst die LEDs in der setup Funktion, die angestoßen wird, sobald der Mikrocontroller Strom erhält, initialisiert und auf dem Display erscheint der Text "Messungen werden gestartet". Danach wird die Loop Funktion, solange der Mikrocontroller Strom hat, durchlaufen. In dieser wird minütlich die Temperatur in der Thermoskanne abgerufen und erscheint auf dem Display. Außerdem wird je nach Temperatur die Farbe der LED gesetzt (siehe Abbildung 5). Diese Farbeinteilung je nach Temperatur ist durch einen Selbsttest entstanden, indem ich überprüft habe, mit welcher Temperatur mir Tee am Besten schmeckt. Für mich hat Tee bei 65 bis 50 Grad die perfekte Temperatur und auch Experten halten 65 bis 60 Grad für die perfekte Trinktemperatur (vgl [6]).

Temperatur	LED Farbe	Kommentar
> 65°C	rot	Zu heiß
65°C – 50°C	grün	Perfekte Temperatur
49°C – 40°C	gelb	Lauwarm
<40°C	blau	Zu kalt

Abbildung 4. Temperaturfarben

5.3. Zusammenspiel

Das Programm, das unter Softwarearchitektur beschrieben wurde, läuft auf dem Wemos D1 mini sobald dieser mit Strom versorgt wird. Der Temperatursensor wird genau wie der LED Streifen mit 3 Kabeln auf ein Steckbrett, auf das der Wemos D1 mini mit dem Display angeschlossen ist, gesteckt. Bei der hier beschriebenen Lösung wurde der Temperatursensor an die Spannungsversorgung (5V und Ground) sowie die Signalleitung (Digital Input D3) und der LED Streifen an die Spannungsversorgung (5V und Ground) sowie die Signalleitung (Digital Input D4) angeschlossen.



Abbildung 5. Realisierung mit Steckbrett

6. Test und Gebrauch

Um die Temperaturanzeige der Thermoskanne zu testen, wurden vier Behälter Tee auf unterschiedliche Temperaturen heruntergekühlt und anschließend in die Thermoskanne gefüllt (siehe Abbildung 7). Durch ein Bratenthermometer konnte überprüft werden, ob die richtige Temperatur angezeigt wird.



Abbildung 6. Use Cases

7. Zusammenfassung

Die größte Herausforderung in diesem Projekt war es, den Mikrocontroller mit der Powerbank unter die Thermoskanne so zu verbauen, dass es technisch einwandfrei funktioniert und optisch gut aussieht. Dabei sollte die Technik jedoch nicht komplett versteckt werden. Aus diesem Grunde habe ich mich dazu entschieden, einen Halter extra für dieses Projekt zu entwerfen und mit dem 3D Drucker auszudrucken.

Eine weitere Schwierigkeit war es, eine für die Idee passende Thermoskanne zu finden. Zuerst ist die Entscheidung auf eine Thermoskanne im Retro Stil gefallen, die meiner Meinung nach noch besser zum Thema Upcycling gepasst hätte. Diese ist jedoch leider für das Verbauen nicht geeignet gewesen und so beim Experimentieren mit der Halterung für den Mikrocontroller zu Schaden gekommen.

Ich nehme aus diesem Seminar die Erfahrung mit, einen (alten) Gegenstand technisch aufzuwerten. Diese Erfahrung kann ich später in der Schule sehr gut nutzen. So plane ich in Zukunft eine Unterrichtsreihe zu diesem Thema durchzuführen, indem die Schülerinnen und Schüler einen von ihnen gewählten Gegenstand mit dem Wemos D1 mini aufbessern sollen. Der Wemos D1 mini bietet dabei den Vorteil, dass dieser in dem vom BMBF geförderten Projekt "Smile" (<https://www.smile-smart-it.de/>), indem ich mitarbeite, mit mBlock blockbasiert programmiert werden und so auch

schon von Programmieranfängern bedient werden kann und ihnen somit einen leichten und erfolgreichen Einstieg in die Programmierung bietet.

Des Weiteren steigert das Aufwerten eines veralteten Gegenstandes die Lernmotivation und Interesse der Schülerinnen und Schüler an Informatik. Denn wahrgenommene inhaltliche Relevanz, sowie Autonomieerleben durch Spielräume im Unterricht stellen Einflussfaktoren auf die Lernmotivation und das Interesse von Schülerinnen und Schüler an Informatik dar. (vgl [7]) Dadurch, dass die Schülerinnen und Schüler am Ende der Unterrichtseinheit einen aufgebesserten Gegenstand präsentieren können, der ihren Alltag erleichtern kann, können sie die inhaltliche Relevanz wahrnehmen und durch das Freistellen des von den Schülerinnen und Schülern ausgewählten Gegenstandes erleben die Schülerinnen und Schüler Autonomie durch Spielräume.

Literatur

- [1] Vorlesungsfolien vom Seminar "Aus alt mach neu: Upcycling mit dem Raspberry Pi" vom 08.April 2019
- [2] GEOlino Lexikon, Band 2, S.1055, Mannheim 2007
- [3] <https://www.wasistwas.de/archiv-technik-details/die-erfinder-der-thermoskanne.html> (Abruf 26.06.2019)
- [4] <https://mwfk.brandenburg.de/sixcms/detail.php/538573> (Abruf 26.06.2019)
- [5] Schüler Duden Die Physik Ein Lexikon der gesamten Schulphysik, S. 71, Mannheim 1974
- [6] <https://www.teekampagne.de/Darjeeling/Zubereitung-Rezepte> (Abruf 26.06.2019)
- [7] Vorlesungsfolien Didaktik der Informatik I: 11.VL: Informatik im Kontext von Ira Diethelm vom 19.Juni 2018

Anhang

Bauteilliste

- Thermoskanne
- LED Streifen mit Kupfer Kabel
- Halter
- Runde Powerbank
- OLED Display
- Wemos D1 mini
- Steckbrett
- USB-Kabel
- Schwarzes Klebeband
- Temperatursensor DS18B20

Code Snippet

```
void setup(){

    //Initialisieren des Displays
    initializeDisplay();

    //Initialisieren der 4 LEDs
    initializeLED();

    //Nach 5 Sekunden beginnen die Messungen
    delay(5000);
    sensors.begin();

}

void initializeDisplay(){

    Serial.begin(115200);
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(32, 9);
    display.println("Messungen");
    display.setCursor(32, 16);
    display.println("werden");
    display.setCursor(32, 23);
    display.println("gestartet");
    display.display();

}

void initializeLED(){

    FastLED.addLeds<NEOPIXEL, LED_DATA_PIN>(leds, NUM_LEDS);
    FastLED.setBrightness(25);
    leds[0] = CRGB(int(255), int(0), int(0));
    leds[1] = CRGB(int(240), int(150), int(15));
    leds[2] = CRGB(int(113), int(226), int(29));
    leds[3] = CRGB(int(0), int(255), int(255));
    leds[4] = CRGB(int(255), int(0), int(0));
    leds[5] = CRGB(int(240), int(150), int(15));
    leds[6] = CRGB(int(113), int(226), int(29));
    leds[7] = CRGB(int(0), int(255), int(255));
    FastLED.show();

}

void loop(){

    //Hier wird der Display einmal geleert, denn sonst würden
    //die Temperaturen immer übereinander geschrieben werden
    display.clearDisplay();

    //Hier wird die Temperatur von dem Sensor abgerufen
    sensors.requestTemperatures();
    double val = sensors.getTempCByIndex(0);

    // Es gibt 4 Farben fuer die Temperatur,
    // 1 steht für rot, also noch zu heiß zum trinken,
    // 2 steht für grün, perfekte Trinktemperatur
    // 3 steht für gelb, nur noch lauwarm
    // 4 steht für blau, dann ist die Flüssigkeit schon zu kalt
    int temperatur = findTemp(round(val));

    //Hier wird die Farbe der LED gesetzt
    setLEDColour(temperatur);

    //Hier wird die Temperatur auf das Display geschrieben
    setTempToDisplay(val);

    //Hier wird eine Sekunde gewartet, damit die Temperatur
    //nur einmal in der Sekunde abgerufen wird
    delay(1000);

}

void setTempToDisplay(double val){

    display.setCursor(32, 12);
    display.println("Temperatur:");

    display.setCursor(32, 22);
    display.println(String(val));

    display.drawCircle(75, 22, 1.8, WHITE);
    display.setCursor(79, 22);
    display.println("C");

    display.display();

}

void setLEDColour(int temperatur){

    for (int i=0; i < NUM_LEDS; i++){

        if (temperatur == 1){
            leds[i] = CRGB(int(255), int(0), int(0));
            FastLED.show();
        }

        else if (temperatur == 2){
            leds[i] = CRGB(int(0), int(255), int(0));
            FastLED.show();
        }

        else if (temperatur == 3){
            leds[i] = CRGB(int(255), int(255), int(0));
            FastLED.show();
        }

        else if (temperatur == 4){
            leds[i] = CRGB(int(0), int(0), int(255));
            FastLED.show();
        }

    }

}

int findTemp(double sensorTemp){

    // > 65 Grad rot
    if (sensorTemp > 65){
        return 1;
    }

    // 65 - 50 Grad grün
    else if(sensorTemp > 50){
        return 2;
    }

    // 49 - 40 Grad gelb
    else if(sensorTemp > 40){
        return 3;
    }

    // < 40 Grad blau
    else {
        return 4;
    }

}

}
```

Die Upcycling Thermoskanne

Jana Gerdes

DIE THERMOSKANNE

Eine normale Thermoskanne

- hält die Temperatur einer Flüssigkeit über einen gewissen Zeitraum konstant
- wurde im 19. Jahrhundert erfunden
- die Bauweise mit einer Doppelwand mit Vakuum ermöglicht das Konstanthalten der Temperatur
- wird am Häufigsten für Tee oder Kaffee genutzt

Die Upcycling Thermoskanne

- Temperaturanzeige der Flüssigkeit in der Thermoskanne durch
 - Display
 - LED Streifen
- verhindert das Verbrennen an zu heißen Flüssigkeiten
- verhindert das Trinken von zu kaltem Tee oder Kaffee



REALISIERUNG

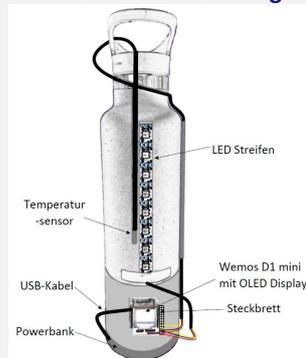
Realisierung

- LED Streifen wurde an der Außenseite der Thermoskanne geklebt
- Wemos D1 mini wurde mit einer Powerbank unterhalb der Thermoskanne verbaut und mit dem LED Streifen und einem Temperatursensor verbunden (siehe Foto)
- in Arduino programmiert leuchtet der LED Streifen je nach vom Temperatursensor gemessener Temperatur (rot, grün, gelb oder blau)

Der Temperatursensor DS18B20

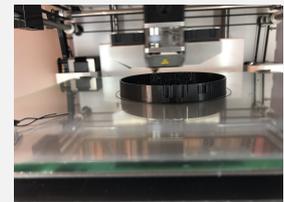
- gelangt über den Trinkstrohalm der Thermoskanne in die Flüssigkeit

Foto der Realisierung



Der Halter

- mit BlocksCAD designt
- mit einem Ultimaker 2+ gedruckt
- macht die technisch und optisch beste Realisierung möglich



USE CASES



Der Inhalt ist zu kalt. Die LED leuchtet blau. Das Display zeigt 37,56 Grad an.



Der Inhalt ist lauwarm. Die LED leuchtet gelb. Das Display zeigt 45,4 Grad an.



Der Inhalt hat die perfekte Temperatur. Die LED leuchtet grün. Das Display zeigt 61,31 Grad an.



Der Inhalt ist zu heiß. Die LED leuchtet rot. Das Display zeigt 85,06 Grad an.

BAUTEILLISTE

Diese Teile werden benötigt:

- Thermoskanne, 10€ (Nr.: 1,2,12,13)
- LED Streifen mit Kupfer Kabel 1€ (Nr.: 3)
- Halter, ca 1€ (Nr.: 4)
- Runde Powerbank, 10€ (Nr.: 5)
- OLED Display, 6€ (Nr.: 6)
- Wemos D1 mini 6€ (Nr.: 7)
- Steckbrett, 1€ (Nr.: 8)
- USB-Kabel, 6€ (Nr.: 9)
- Schwarzes Klebeband 3€ (Nr.: 10)
- Temperatursensor DS18B20, 2,50€ (Nr.: 11)



DIE TEMPERATURBEREICHE

Temperatur	LED Farbe	Kommentar
> 65°C	rot	Zu heiß
65°C – 50°C	grün	Perfekte Temperatur
49°C – 40°C	gelb	Lauwarm
<40°C	blau	Zu kalt

LITERATUR

Vorlesungsfolien vom Seminar "Aus alt mach neu: Upcycling mit dem Raspberry Pi" vom 08. April 2019; GEOlino Lexikon, Band 2, S.1055. Mannheim 2007; <https://www.wasistwas.de/archiv-technik-details/die-erfinder-der-thermoskanne.html> (Abruf 26.06.2019); <https://mwfk.brandenburg.de/sixcms/detail.php/538573> (Abruf 26.06.2019); Schüler Duden Die Physik Ein Lexikon der gesamten Schulphysik, S. 71, Mannheim 1974; <https://www.teekampagne.de/Darjeling/Zubereitung-Rezepte> (Abruf 26.06.2019); Vorlesungsfolien Didaktik der Informatik I: 11.VL: Informatik im Kontext von Ira Diethelm vom 19. Juni 2018;

DIY-Longboard mit Gewichtverlagerung

Dimitri Manschos
Wirtschaftsinformatik
Carl von Ossietzky Universität
Oldenburg

Abstract

In dieser Arbeit geht es darum, dass ein bestehendes Produkt upgecyclet wird und man somit dieses Produkt zu ein höherwertiges Produkt entwickelt. Das Produkt, um welches es in dieser Ausarbeitung handelt, ist das Longboard. Das Longboard soll die Eigenschaft der Gewichtverlagerung des derzeit beliebten Hoverboards verwenden. Hierzu werden Wägezellen verwendet, welche auf Gewichtunterschiede reagieren. Sobald ein Gewichtunterschied entsteht, wird ein Signal, über eine Motorsteuerung, an ein elektronischen Motor gesendet. Um die Rotation des Motors auf die Reifen des Longboards zu übertragen, wird ein Riemensystem verwendet. Der Motor wird mithilfe eines LiPo-Akkus betrieben. Es wird kein Raspberry Pi verwendet, da ein PWM-Signal benötigt wird, um ein ESC-Motor ansteuern zu können. Da dieses PWM-Signal mithilfe eines Arduino gesendet werden kann, wird das Projekt mithilfe eines Arduino Uno realisiert.

1. Einleitung

Um über das Longboard etwas sagen zu können, muss zunächst genauer auf das Skateboard eingegangen werden. Das Skateboard ist ein Fortbewegungsmittel, mit welchem gefahren, als auch getrickt werden kann. Beim Fahren wird ein Bein auf das Board gestellt, während das andere für die Beschleunigung zuständig ist. Sobald eine optimale Geschwindigkeit erreicht wurde, sind beide Beine auf dem Board. Wird nun das Gewicht vertikal verlagert, so bewegen sich die Achsen zueinander und es wird eine Steuerung des Boardes möglich. Das Tricksen mit dem Board ist sowohl während der Fahrt, als auch im Stand möglich. Hierbei entsteht eine große Variation von verschiedenen Tricks, welche mit dem Skateboard ausgeführt werden können. Hierzu spielt oftmals die Beinstellung eine entscheidende Rolle, welcher Trick am Ende entsteht. Das Longboard entstand aus der Idee des Skateboards, mit dem Unterschied, dass es beim Longboard weniger um das Tricksen geht. Das

Tricksen wurde somit größtenteils entfernt und die Funktion des Fahrens wurde verbessert. Bei einem Longboard werden größere Rollen verwendet, womit auch eine höhere Geschwindigkeit erreicht werden kann.

1.1. verschiedene Möglichkeiten

Heutzutage sieht man viele Personen auf der Straße, welche sich mit einem Hoverboard fortbewegen. Das Prinzip hierbei ist, dass die Person, welche das Hoverboard verwendet, ihr eigenes Körpergewicht verlagern muss, so dass sich das Hoverboard in die gewünschte Richtung bewegt. Um diesem Trend entgegenzuwirken dachte ich mir, dass man ein älteres, freizeitleiches Fortbewegungsmittel upcyclen könnte. Unzwar das Skateboard. Hierbei ergeben sich zwei unterschiedliche Möglichkeiten das Skateboard fortzubewegen. Die erste Idee ist es Drucksensoren zwischen Achse und Board zu befestigen. Dies müsste jeweils auf der vorderen- und der hinteren Seite des Board befestigt werden. Sobald mehr Druck auf der vorderen Achse des Boards entsteht, sollen zwei elektronische Motoren, über Zahnräder die Rollen in die vordere Richtung bewegen. Umso höher der Druck, desto schneller soll das Board werden. Das ganze könnte optional über ein Display, welches auf der oberen Seite des Skateboards eingebaut ist, angezeigt werden. Die zweite Möglichkeit wäre identisch, jedoch würden keine Drucksensoren verwendet werden, sondern Sensoren. Hierbei würde man kleine Löcher auf die vordere Seite des Skateboards bohren um die Sensoren in diesen Löchern zu befestigen. Sobald ein Fuß ein Sensor verdeckt, sollen die elektronischen Motoren wie in Möglichkeit eins arbeiten. Sobald die Sensor, welcher weiter vorne liegt mit dem Fuß verdeckt werden, wird die Geschwindigkeit erhöht. Also wurde in dieser Möglichkeit Stufenweise beschleunigt.

Zunächst wurde sich in diesem Projekt wurde für die erste Variante entschieden, jedoch wird anstelle eines Skateboards, ein Longboard verwendet. Die Elektronik wäre bei Tricks und Sprüngen mit dem Skateboard leicht zu beschädigen. Außerdem wurde sich zur

Kraftübertragung des Motors auf den Reifen, für ein Riemensystem entschieden, welches ebenfalls nicht schnell bei Verwendung eines Skateboards abmontierbar wäre, sodass schadenfreies tricksen möglich wäre. Aus den genannten Gründen habe ich mich gegen meinen ersten Einfall des Skateboards und für die Idee des Longboards entschieden.

Im Endeffekt wurde Methode eins überarbeitet, da ein HX711 nicht die Werte zweier Wägezellen individuell auslesen kann. Hierfür müssten zwei Wägezellen verwendet werden, welche wiederum jeweils an einem HX711 angeschlossen werden müssten. Nun wurden zwei Wägezellen auf der vorderen Seite platziert, sodass das Gewicht der beiden Wägezellen addiert und so mit einem genaueren Wert gearbeitet werden kann. Sobald eine Person auf das elektronische Skateboard steigt, wird ein Wert des Gewichts gespeichert. Hierzu muss sich der Benutzer neutral auf das Longboard stellen, wobei das Board eben platziert werden muss. Sobald das momentane Gewicht den gespeicherten Wert übersteigt, wird das Board beschleunigt. Ist der momentane Wert jedoch viel geringer, wird das Board abgebremst. Außerdem wurde aus Kostengründen nur ein Motor verwendet.

2. mechanischer Teil

Zunächst musste das Longboard bearbeitet werden, da der Abstand von Board zu Boden zu gering war und an dieser Stelle ein Motor eingebaut werden musste. Da die Achsen über eine kleine Metalplatte auf dem Boarddeck befestigt war, musste zunächst die Achse in ihre einzelnen Komponenten auseinander gebaut werden. Anschließend wurden unebenheiten des Holzes mit Schleifpapier korrigiert und die Achse wieder zusammengebaut. Diese wurde dann auf der unteren Seite des Boards befestigt, sodass genug Platz für den Motoren existiert. Da nur ein Motor verwendet wird, musste eine Rolle zunächst abmontiert werden. In diese Rolle wurden Löcher gebohrt, sodass hier ein Zahnriemen befestigt werden konnte. Sodass eine Rolle mit dem befestigten Zahnriemen auf die Achse passte, musste die Achse auf dieser Seite geflext werden. Zuletzt musste der Motor befestigt, sowie die Wägezellen optimal platziert werden, sodass ein Druckunterschied erkannt werden kann.

3. elektronische Verbindung

Verwendete Teile:

Arduino-Uno

ESC-Motor

ESC-Motorsteuerung

LiPo-Akku

HX711

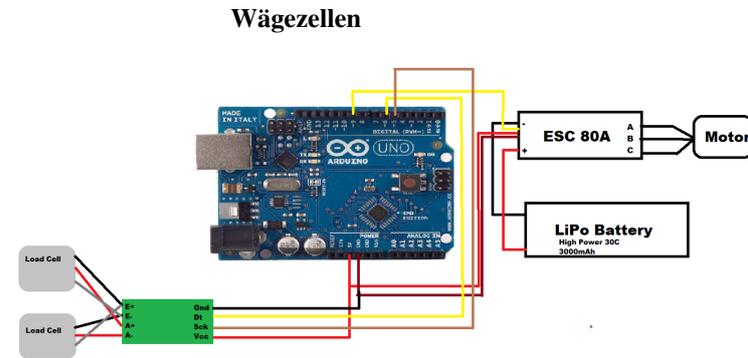


Figure 1. Verbindungen
[1] und [2]

Verbindung ESC-Motor und ESC-Motorsteuerung

Die A,B und C Schnittstellen sind mit dem Motor verbunden, dabei bestimmt die Reihenfolge in welche Richtung sich der Motor dreht.

Verbindung ESC-Motor und LiPo-Akku Plus und Minus sind miteinander verbunden, sodass der Motor den Strom von dem Akku bezieht.

Verbindung ESC-Motorsteuerung und Arduino Uno

Signal ist über die Schnittstelle Digital Port 9 verbunden. Über diesen Port wird der Motor gesteuert und das senden eines PWM-Signals ist möglich. Die 5V Schnittstelle wird mit der 5V Schnittstelle des Arduinos verbunden und die beiden GND Schnittstellen werden miteinander verbunden.

Verbindung Arduino-Uno und HX711 Die beiden GND Schnittstellen werden miteinander verbunden. Außerdem wird die VCC Schnittstelle mit der 5V Schnittstelle des Arduino-Unos verbunden. DT zu Digital 6 und Sck zu Digital 5, sodass eine Ansteuerung der angeschlossenen Wägezellen möglich ist.

Verbindung HX711 und Wägezellen Wenn eine Wägezelle verwendet wird muss diese anders verbunden werden. Jedoch wurde in diesem Projekt zwei Wägezellen verwendet. Hierbei muss der Schwarze Kabel der ersten Wägezelle und der weiße Kabel der zweiten mit der Schnittstelle E+ des HX711 verbunden werden. Das gleiche gilt für die Schnittstelle E-, nur sind die Kabel der ersten und zweiten Wägezelle vertauscht. Anschließend werden der rote Kabel der ersten Wägezelle mit A+, sowie der rote Kabel der zweiten Wägezelle mit A- verbunden.

4. Ablauf

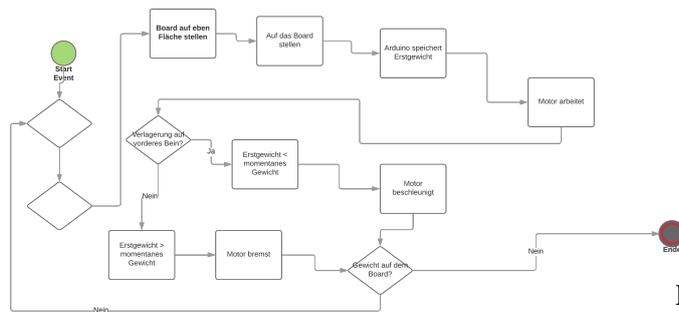


Figure 2. Verbindungen

```
if(ersterwert <= momentaner Wert )
{
    Serial.println("fahren");
}
else if(ersterwert >= momentaner Wert )
{
    Serial.println("stop");
}
```

Figure 3. Beispiel

Zuerst muss der Benutzer das Board auf eine eben Fläche stellen, sodass der Erstgewicht-Wert nicht verfälscht wird, sobald sich die Person auf das Board stellt. Sobald der erste Wert gespeichert wurde, fängt der Motor an zu arbeiten. Nun muss der Benutzer mit dem Board agieren. Wird mehr Gewicht auf das vordere Bein verlagert, so wird der erste Wert mit dem momentanen Gewicht, welches auf der vorderen Achse lastet verglichen und bringt den Motor dazu zu beschleunigen. Wird das Gewicht nach hinten verlagert, so ist der erste Wert größer, als der Momentane und der Motor bremsst die Geschwindigkeit ab.

5. Erweiterungsmöglichkeiten

Motor Zu einem könnten bessere Motoren verwendet werden, welche mehr Leistung erbringen. Jedoch wäre der Einsatz von zwei Motoren auch möglich.

LiPo-Akku Zwei LiPo-Akkus würden die Laufzeit des Longboardes erhöhen.

HX711 Mit zwei HX711 könnten beide Wägezellen individuell angesteuert werden.

Danksagung

Mein Dank gilt Marvin Banse, welcher mir bei jeder Frage behilflich war.

References

- [1] Dejan, "Arduino Brushless Motor Control Tutorial — ESC — BLDC," unter: <https://howtomechatronics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-bldc/> (abgerufen am 02.07.2019)
- [2] mybotic, "Tutorial to Interface HX711 Balance Module With Load Cell," unter: <https://www.instructables.com/id/How-to-Interface-HX711-Balance-Module-With-Load-Ce/> (abgerufen am 01.07.2019)

DIY-Longboard steuern durch Gewichtsverlagerung

Dimitri Manschos

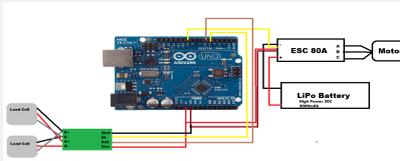
MOTIVATION DES PROJEKTES

- DIY-Longboard, welches mithilfe von Gewichtsverlagerung betrieben wird, anstelle der häufig verwendeten Fernsteuerung
 - Gewichtsverlagerung wird nur in den "Hoverboards" verwendet
 - Diese Technologie nun in einem Longboard verwenden
1. Ein "Boosted Board" ist sehr preisintensiv
 2. bisherige elektronischen Boards verwenden Fernsteuerung
 3. elektronische Boards mit Gewichtsverlagerung zu beschleunigen noch nicht vorhanden



PROZESS: SCHRITTANLEITUNG - ABLAUF

- Schritt 1: Verbindung der elektronischen Teile



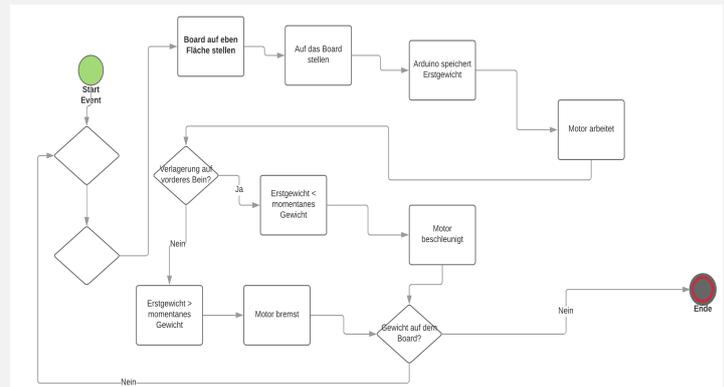
- Schritt 2: Platzierung der Wägezellen unter der vorderen Achse



- Schritt 3: Motorhalterung an die hintere Achse befestigen
- Schritt 4: Motor an Motorhalterung befestigen
- Schritt 5: Zahnrad an die Rolle montieren
- Schritt 6: Zahnriemen zwischen Zahnradern spannen
- Schritt 7: Elektronik verstauen und an die unterseite des Boards anbringen

- Funktionsweise

- Sobald das Board betreten wird, wird Erstgewicht gespeichert
- $\text{Erstgewicht} < \text{momentanes Gewicht} = \text{Motor beschleunigt das Board}$
- $\text{Erstgewicht} > \text{momentanes Gewicht} = \text{Motor bremst das Board}$



ENDERGEBNISSE

- **elektronisches Longboard**



- **Eckdaten**

- Länge: 97 cm
- Breite: 22 cm
- max. Gewicht: ca. 100 kg
- Abstand Verstaubox-Boden: 4,2 cm
- Preis: ca. 190 €

- **Leistungsdaten**

- Akkulaufzeit: ca. 30 min
- max. Geschwindigkeit: ca. 17 km/h

ERWEITERUNGSMÖGLICHKEITEN

- **technisch**
 - LEDs anbringen
 - Motor in beide Richtungen antreiben
 - 2*HX711 verwenden, sodass Wägezellen individuell angesteuert werden
- **Komponenten**
 - Motor mit mehr Leistung verwenden bzw. mehrere Motoren
 - Akku mit höherer Laufzeit verwenden bzw. mehrere Akkus parallel schalten

VERWENDETE QUELLEN

"Arduino Brushless Motor Control Tutorial — ESC — BLDC," unter: <https://howtomechatronics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-bldc/> (abgerufen am 02.07.2019)

"Tutorial to Interface HX711 Balance Module With Load Cell," unter: <https://www.instructables.com/id/How-to-Interface-HX711-Balance-Module-With-Load-Cell/> (abgerufen am 01.07.2019)

Kaminuhr 2.0

Upcycling einer Kaminuhr

Florian Schmalriede, B.Sc. Informatik
Systemsoftware und verteilte Systeme
Carl von Ossietzky Universität Oldenburg

Abstract

Im Rahmen dieser Arbeit wird das Upcycling einer Kaminuhr beschrieben. Um die Nachteile eines mechanischen Uhrwerks zu überwinden, wurde dieses durch ein Mikrocontroller-gesteuertes Uhrwerk ersetzt. Durch entsprechender Software war es so möglich, den Funktionsumfang der Uhr zu erweitern. Die resultierende Kaminuhr verfügt über eine Web-Oberfläche zur Steuerung und Konfiguration der Uhr, einer automatischen Zeitstellung, einer Zeitmanipulationsmöglichkeit und einer Chaotischen Zeitanzeige.

1. Einleitung

Das Upcycling von Gegenständen kann diese vor einer Entsorgung retten und fördert so die Sparsamkeit mit Ressourcen. Motiviert durch das Seminar „Aus alt mach neu: Upcycling mit Raspberry Pi und Co.“ der Abteilung Systemsoftware und verteilte Systeme behandelt diese Arbeit das Upcycling einer defekten Kaminuhr. Anfänglich wird die Analyse der Ausgangskaminuhr inklusive potentieller Upcycling-Möglichkeiten beschrieben. Darauf aufbauend folgen die Planung und die Beschreibung der Umsetzung des Upcyclings. Abschließend wird als Fazit gezogen, dass die Kaminuhr vor der Entsorgung gerettet wurde.

1.1. Basis der Kaminuhr 2.0

Die Grundlage der Kaminuhr 2.0 bildet eine Kaminuhr (Abbildung 1) aus den 1960er¹ Jahren. Aufgrund der eingeschränkten Funktionstüchtigkeit war die Kaminuhr für die Entsorgung vorgesehen. Daher kann die Kaminuhr ohne Einschränkungen modifiziert werden. Die Kaminuhr war rein mechanisch und bietet genügend Raum für den Einbau von elektronischen Komponenten. Diese Eigenschaften sorgen für viele Upcycling-Möglichkeiten und führten dazu,

¹Nach Aussage der Großeltern

dass die Kaminuhr als Basis für die Kaminuhr 2.0 verwendet wurde.



Abbildung 1. Basis der Kaminuhr 2.0

2. Analyse

Aufgrund von fehlenden Kennzeichnungen konnten weder Hersteller noch Modell der Kaminuhr ermittelt werden. Ursprünglich konnte die Kaminuhr die Uhrzeit anzeigen und zu festen, nicht konfigurierbaren Uhrzeiten der Stunde entsprechend viele Töne erzeugen.

2.1. Aufbau und Funktionsweise

Die Kaminuhr lässt sich in die vier Baugruppen Gehäuse, Ziffernblatt mit Schutzscheibe, Uhrwerk und Tonstangen unterteilen. Das Uhrwerk und die Tonstangen sind im Gehäuse so verbaut, dass am Uhrwerk befestigte Hämmer die Tonstangen anschlagen können. Das Ziffernblatt mit der Schutzscheibe ist frontal am Gehäuse befestigt. Durch ein Loch im Ziffernblatt und passend dazu im Gehäuse sind die Zeigerwellen für den Stunden- sowie Minutenzeiger geführt. Drei Aufzugwellen des Uhrwerks lassen sich durch drei weitere Löcher im Ziffernblatt mit passenden Löchern im Gehäuse erreichen.

Mit einem entsprechenden Schlüssel lässt sich das Uhrwerk über die Aufzugwellen aufziehen. Die angelegte Energie wird von Federn gespeichert und zum Antrieb des Uhr-

werks und somit der Uhrzeiger verwendet. Zur vollen Stunde wird ein Teil der angelegten Energie für die Tonerzeugung durch die Hämmer verwendet. Nachdem die komplette Energie aus dem Federspeicher umgewandelt wurde, bleibt die Uhr stehen und muss erneut aufgezogen werden.

2.2. Upcycling-Möglichkeiten

Das mechanische Uhrwerk erfordert ein regelmäßiges Aufziehen. In der heutigen elektrifizierten Welt wird diese Tätigkeit häufig als lästig empfunden. Auch schränkt die mechanische Bauweise den Funktionsumfang der Kaminuhr ein. Viele Geräte können heutzutage mit einem Softwareupdate um weitere Funktionen ergänzt werden. Mit der Ersetzung des mechanischen Uhrwerks durch ein Mikrocontroller, der als Uhrwerk dient, sollen diese Nachteile überwunden werden. Zusätzlich erlaubt der Einsatz eines Mikrocontrollers in Kombination mit geeigneter Software und Hardware noch weitere Upcycling-Möglichkeiten. Im folgenden sind einige Ideen hierzu aufgelistet.

Getrennte Ansteuerung von Stunden- und Minutenzeiger Für die mechanische Stellung von Stunden- und Minutenzeigerwelle eignet sich ein Schrittmotor. Dieser kann mit einem Mikrocontroller entsprechend der jeweiligen Uhrzeit angesteuert werden. Durch den Einsatz von zwei Schrittmotoren kann die Stundenzeigerwelle unabhängig von der Minutenzeigerwelle positioniert werden. Dies beschleunigt die Umstellung der Anzeige und erlaubt neue Anzeigemöglichkeiten. Neben Uhrzeit können auch andere Informationen, welche durch zwei Zahlen oder Ausrichtung darstellbar sind, angezeigt werden. Als Beispiel wäre die aktuelle Temperatur denkbar. Auch erlaubt die getrennte Ansteuerung von Stunden- und Minutenzeiger die Realisierung einer chaotischen Uhr.

Zeitsynchronisation Die derzeit noch gültige Zeitumstellung erfordert bei dem mechanischen Uhrwerk zweimal im Jahr eine manuelle Umstellung der Zeiger. Aufbauend auf einem Mikrocontroller-gesteuertem Uhrwerk kann eine Zeitsynchronisation realisiert werden. Diese sorgt dafür, dass eine manuelle Umstellung der Zeiger nicht mehr notwendig ist. Außerdem werden Laufzeitfehler durch eine Zeitsynchronisation wiederholt korrigiert. So ergibt sich auch bei einer längeren Laufzeit der Uhr eine nur geringe Abweichung zur Zeit.

Benutzungsschnittstelle Eine integrierte Benutzungsschnittstelle erlaubt die Konfiguration der Uhr. So kann diese dem individuellen Bedürfnissen angepasst werden. Wenn die Benutzungsschnittstelle zusätzlich über ein Netzwerk erreichbar ist, dann kann der jeweilige Benutzer die Uhr auch aus der Ferne konfigurieren.

Konfigurierbare Töne Die Variation der „Melodien“, welche von der Kaminuhr erzeugt werden können, ist sehr eingeschränkt. Durch die Ergänzung von Lautsprechern und entsprechende Software zum Abspielen und Konfigurieren von Melodien könnte der Besitzer individuelle Klänge konfigurieren und abspielen lassen.

Wecker Aufbauend auf den konfigurierbaren Tönen ist auch eine Konfiguration der Zeitpunkte, an denen diese abgespielt werden sollen, möglich. Die bisherige Kaminuhr war aufgrund der mechanischen Bauweise nur zu festen Zeitpunkten in der Lage die Töne zu spielen. Mit der freien Wahl von Zeitpunkten, an denen Töne abgespielt werden, wäre eine Wecker-Funktion gegeben. Wenn die Uhr mit dem Internet verbunden wird, dann wäre sogar die Funktion eines Radio-Weckers möglich. Über die Benutzungsschnittstelle könnte der Besitzer seinen favorisierten Internetradio-Dienst angeben.

Beleuchtung Bei schwacher Beleuchtung ist es derzeit nur schwer bis gar nicht möglich, die Uhrzeit auf der Kaminuhr abzulesen. Mit einer entsprechenden Beleuchtung könnte auch ohne externe Lichtquelle die Uhrzeit immer sichtbar sein. Zusammen mit den Ansteuerungsmöglichkeiten eines Mikrocontrollers ergeben sich zusätzlich Variationen in der Beleuchtung. Über die Benutzungsschnittstelle könnte der Besitzer seine favorisierten Farben einstellen. Auch ein zeitlich wechselnder Farbverlauf ist denkbar.

Zeitmanipulation Bei erfreulichem Besuch vergeht die Zeit meist viel zu schnell und bei lästigen Besuch oft zu langsam. Der Mikrocontroller in Kombination mit einer aus der Ferne erreichbaren Benutzungsschnittstelle könnte den Benutzer die Manipulation der Zeit erlauben. Ein konfigurierbarer Manipulationsfaktor würde dafür sorgen, dass die Uhr schneller oder langsamer läuft. Die Fernkonfiguration erlaubt den Benutzer diese Konfiguration unbemerkt vorzunehmen. Dem Besuch wird so eine falsche Uhrzeit vorgebracht und er verweilt dementsprechend länger oder kürzer. Sobald der Besuch beendet ist, könnte die richtige Uhrzeit über die Benutzungsschnittstelle wieder hergestellt werden.

3. Kaminuhr 2.0

Wie in Abschnitt 2.2 aufgezeigt, sind viele Upcycling-Möglichkeiten für die Kaminuhr gegeben. Die Kaminuhr 2.0 soll hiervon die getrennte Ansteuerung von Stunden- und Minutenzeiger, die Zeitsynchronisation, die Benutzungsschnittstelle und die Zeitmanipulation vereinen.

Eine über ein Netzwerk erreichbare Web-Oberfläche soll als Benutzungsschnittstelle dienen und es dem Besitzer erlauben, eine zeitgemäß einfache Konfiguration der Uhr

über ein Smartphone oder einen PC vorzunehmen. Über die Web-Oberfläche soll zwischen den vier Betriebsmodi *Manuell*, *NTP*, *Time-Travel* und *Chaos* gewählt werden können. Im Betriebsmodus *Manuell* kann die Uhr manuell über die Web-Oberfläche eingestellt werden. Der Mikrocontroller passt fortlaufend die Uhrzeit entsprechend an, so dass die Uhrzeit relativ zur eingestellten Uhrzeit stimmig ist. Um eine manuelle Einstellung zu vermeiden, kann der Betriebsmodus *NTP* gewählt werden. In diesem Modus wird eine Referenzzeit in konfigurierbaren Zeitabständen von einem beliebigen NTP-Server abgefragt und als Uhrzeit übernommen. Zwischen den Abfragen sorgt der Mikrocontroller für eine fortlaufende Uhrzeit. Die Zeitmanipulation wird mit dem Betriebsmodus *Time-Travel* umgesetzt. Dieser lässt wie der *Manuell*-Betriebsmodus die manuelle Stellung der Uhrzeit zu. Weiter kann der Benutzer über die Web-Oberfläche einen Manipulationsfaktor bestimmen, welcher im *Time-Travel*-Betriebsmodus dazu genutzt wird, die Zeit schneller bzw. langsamer laufen zu lassen. Wenn der Betriebsmodus *Time-Travel* verlassen wird, dann wird die Anfangsuhrzeit automatisch wiederhergestellt. Der Betriebsmodus *Chaos* lässt die Uhr chaotisch zwischen Uhrzeiten springen. Hierzu werden fortlaufend zufällige Uhrzeiten erzeugt und durch die Uhrzeiger angesteuert. Die Zeiger drehen sich dabei zufällig im Uhrzeigersinn bzw. gegen den Uhrzeigersinn.

3.1. Verwandte Arbeiten

Ähnlich zum *Chaos*-Betriebsmodus verhält sich die Uhr aus dem Projekt *Time is Fake* [1]. Die in diesem Kunstprojekt entstandene Uhr verhält sich vollkommen willkürlich und steuert Stunden- sowie Minutenzeiger ebenfalls getrennt an. Dieses Verhalten soll Betrachter auf amüsante Weise dazu einladen das Konzept der Zeit zu hinterfragen. Durch einen Handtaster lässt sich das chaotische Verhalten der Uhr für sieben Sekunden unterbrechen. Während dieser Unterbrechung wird die richtige Ortszeit angezeigt. Die Uhr kann im Hinterhof der Kulturetage in Oldenburg betrachtet werden. Normalerweise wird auch eine Live-Übertragung der Uhr über eine Webseite angeboten. Diese war jedoch zum Abrufzeitpunkt nicht gegeben.

4. Entwurf

Um die gewünschten Funktionalitäten umzusetzen, soll das Uhrwerk entfernt und durch neue Komponenten ersetzt werden. In Abbildung 2 ist das neue System und dessen Hardwarekomponenten dargestellt. Über ein Access Point verbindet sich der Mikrocontroller drahtlos mit einem Netzwerk. Ein im Netzwerk befindliches Endgerät soll so in der Lage sein, mit dem Mikrocontroller zu kommunizieren. Über digitale Ausgangspins kommuniziert der Mikro-

controller mit jeweils einem Schrittmotortreiber für die Ansteuerung von Minuten- bzw. Stundenzeiger. Die Schrittmotortreiber setzen die digitalen Schrittbefehle in analoge um und steuern damit über eine Verbindung die Schrittmotoren für Minuten- und Stundenzeiger an. Durch eine physikalische Verbindung zwischen Schrittmotorwelle und Minuten- bzw. Stundenzeiger werden die Drehungen von den Schrittmotoren auf diesen umgesetzt. Bei der Verbindung zwischen Schrittmotorwelle und Minuten- bzw. Stundenzeiger muss darauf geachtet werden, dass mindestens 60 Schritte pro Umdrehung notwendig sind. Wenn weniger Schritte pro Umdrehung benötigt werden, dann kann nicht jede Minute einzeln angezeigt werden.

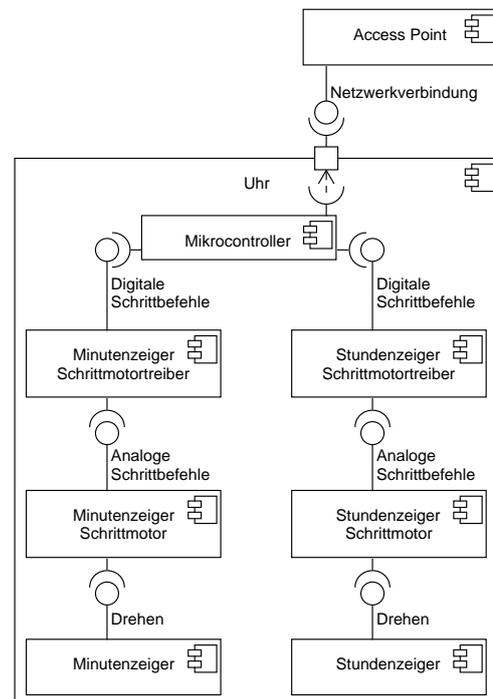


Abbildung 2. Hardwarekomponenten

Der Mikrocontroller ist die zentrale Steuereinheit und kümmert sich um den Aufbau einer Netzwerkverbindung, die Uhrzeitberechnung, die Uhrzeitanzeige, die Konfigurationsverwaltung, die Abbildung von Betriebsmodi und die Bereitstellung einer Benutzungsschnittstelle. Damit das Mikrocontroller-Programm wartbar und erweiterbar bleibt, wurde dieses in Softwarekomponenten aufgeteilt. Jeder Komponente entfallen wenige Aufgaben, womit diese eine geringe Komplexität aufweisen. In Abbildung 3 werden die einzelnen Softwarekomponenten dargestellt.

Einige Komponenten im Programm lassen sich durch den Benutzer konfigurieren. Damit diese Konfigurationen selbst bei einem Verlust der Stromquelle beibehalten wer-

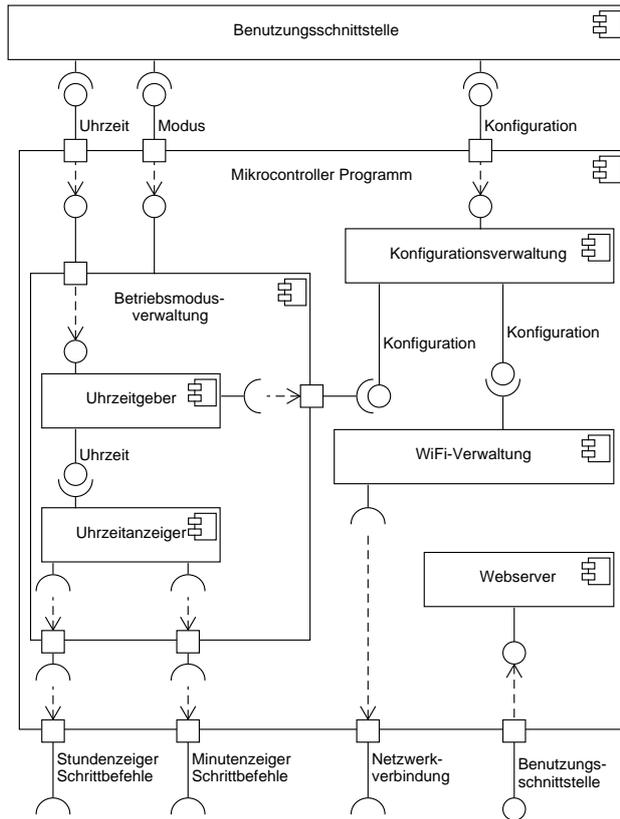


Abbildung 3. Softwarekomponenten

den, müssen diese fest gespeichert werden. Die Konfigurationsverwaltung übernimmt diese Aufgabe und stellt Schnittstellen für anderen Komponenten bereit, damit diese Konfigurationen setzen bzw. lesen können.

Den Aufbau einer Netzwerkverbindung wird von der WiFi-Verwaltung übernommen. Anhand der Konfiguration aus der Konfigurationsverwaltung entscheidet die Komponente, ob ein eigener Access Point erstellt oder eine Verbindung zu einem bestehenden Access Point aufgebaut werden soll. Sollte keine Verbindung zu einem konfigurierten Access Point möglich sein, startet die Komponente wiederum einen eigenen Access Point.

Der Webserver stellt die Benutzungschnittstelle über HTTP für Endgeräte bereit. Außerdem fungiert der Webserver als Kommunikationsbrücke zwischen der Benutzungschnittstelle und anderen Komponenten.

Vom Webserver wird die Benutzungschnittstelle in Form einer Webanwendung ausgeliefert. Die Benutzungschnittstelle erlaubt den Benutzer grafisch mit dem System zu interagieren. Für die Interaktion nutzt die Benutzungschnittstelle Schnittstellen von anderen Komponenten aus dem System.

Die Verwaltung von Betriebsmodi wird durch die Modusverwaltung realisiert. Anhand des jeweiligen Modus entscheidet die Modusverwaltung, welcher Uhrzeitgeber und welcher Uhrzeitanzeiger verwendet wird. Der Wechsel des Betriebsmodus soll dabei im Betrieb möglich sein.

Vom Uhrzeitgeber wird die aktuelle Uhrzeit berechnet und über Schnittstellen den anderen Komponenten bereitgestellt. Je nach Betriebsmodus kann sich der Uhrzeitgeber ändern. So kann zum Beispiel zwischen einer manuell gestellten Uhr zu einer automatisch gestellten Uhr gewechselt werden.

Der Uhrzeitanzeiger kümmert sich um die Anzeige der aktuell berechneten Uhrzeit. Wie der Uhrzeitgeber kann diese Komponenten während des Betrieb der Uhr ausgetauscht werden. So ist es möglich zwischen der Anzeige der aktuellen Uhrzeit und einem Kalibrierungsmodus, zum Kalibrieren der Zeiger, zu wechseln.

5. Realisierung

Die Hardware wurde, wie in Abbildung 4 dargestellt, auf Basis eines Breadboards aufgebaut. Das Breadboard passt zusammen mit den Hardwarekomponenten in das Gehäuse der Kaminuhr und ermöglicht eine prototypische Entwicklung. Löten ist durch das Breadboard überflüssig und alle verbauten Komponenten können leicht durch andere ersetzt werden.

Als Mikrocontrollerboard wurde das NodeMCU Lua Amica Modul V2 ESP8266 ESP-12E gewählt. Dieses wird günstig angeboten und der ESP8266 Mikrocontroller bietet die gewünschten Funktionalitäten. Im Mikrocontroller ist ein WiFi-Modul integriert und auf dem Mikrocontrollerboard eine WiFi-Antenne angebracht. Über Software-Bibliotheken lässt sich das WiFi-Modul direkt ansteuern. Damit ist kein externes Modul für die drahtlose Netzwerkverbindung notwendig. Der ESP8266 verfügt außerdem über einen Flash-Speicher welcher die feste Speicherung von Dateien und Konfigurationen erlaubt. Durch Software-Bibliotheken können Dateien über SPIFFS (Serial Peripheral Interface Flash File System) geschrieben oder gelesen werden. Ein Plug-In für das Arduino-Studio erlaubt sogar das direkte kopieren von Dateien in das Dateisystem des Mikrocontrollers. Dies erleichtert das Aufspielen der Dateien für die Benutzungschnittstelle. Neben dem SPIFFS kann der Flashspeicher auch über eine EEPROM-Emulation angesprochen werden. Bestehende Software-Bibliotheken setzen darauf auf und erlauben das Serialisieren von Daten in den Speicher. Der ESP8266 verfügt zwar über relativ viele digitale Ausgabepins allerdings werden viele für interne Funktionen verwendet, womit die Anzahl der frei belegbaren Pins stark reduziert wird. Jedoch bietet der Mikrocontroller eine I²C Schnittstelle. Über diese können IO-Expander zur Erweiterung der Ausgabepins angeschlossen

werden.

Aufgrund von mangelnden Ausgangspins wurde die I²C Schnittstelle des Mikrocontrollers genutzt um einen Laqiya MCP23017-E anzuschließen. Dieser IO-Expander erweitert die Anzahl der Ausgangspins um 16. So ist es möglich mit dem ESP8266 zwei Schrittmotoren vom vorliegenden Typ anzusteuern ohne auf interne Funktionalitäten des Mikrocontrollers zu verzichten.

Als Schrittmotortreiber wurde der ULN2003 ausgewählt. Diesen gibt es preisgünstig im Set zusammen mit dem 28BYJ-48 Schrittmotor. Der niedrige Preis und die einfache Ansteuerung der Motoren über den ULN2003 Schrittmotortreiber waren ausschlaggebend für diese Wahl. Jedoch hat der 28BYJ-48 Schrittmotor eine Schrittübersetzung von 1/32 welche nicht für die Darstellung von Minuten ausreicht. Allerdings ist im 28BYJ-48 Schrittmotor ein Getriebe integriert, welches eine Untersetzung hat und somit die notwendige Übersetzung von mindestens 1/60 Schritte pro Umdrehung gewährleistet. Eine eindeutige Kennzeichnung der Übersetzung des Getriebes konnte nicht ermittelt werden. Daher wurde einer der bestellten Schrittmotoren geöffnet und die Übersetzungen gezählt. Bei dem eingesetzten Modell liegt eine Übersetzung von 1/64 vor. Somit ergeben sich 2048 Schritte pro Umdrehung welche für die Darstellung von Minuten ausreichend sind.

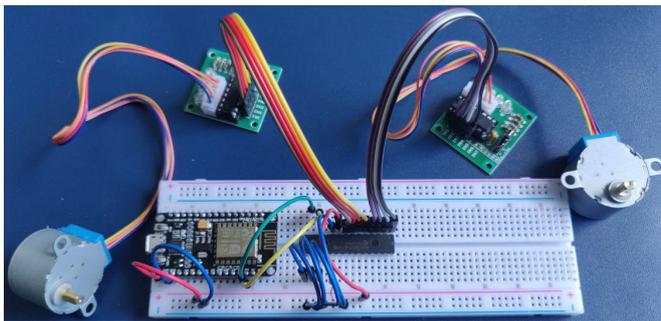


Abbildung 4. Realisierung der Hardware

Es war günstiger die Schrittmotortreiber, Schrittmotoren und IO-Expander jeweils in einem Set zu bestellen. Daher stimmen die Mengenangaben in der Tabelle 1 nicht mit den tatsächlich benötigten Bauteilen überein. Die Gesamtkosten für nur benötigte Bauteile beträgt 21,73 €.

Bauteil	Menge	Kosten in €
Breadboard inkl. Kabel	1	6,99
Schrittmotor inkl. Treiber	5	14,38
Mikrocontrollerboard	1	5,99
IO-Expander	5	14,99

Tabelle 1. Kosten der Bauteile

Der ESP8266 Mikrocontroller lässt sich über das Arduino-Studio programmieren. Wie in diesen Programmen üblich, wird erst eine *setup*-Routine und danach wiederholend eine *loop*-Routine aufgerufen. In Abbildung 5 ist das Mikrocontroller Programm durch ein Aktivitätsdiagramm zusammengefasst und lässt die *setup*- sowie *loop*-Routine erkennen.

In der *setup*-Routine wird erst die Konfiguration aus dem Flashspeicher geladen. Entsprechend der Konfiguration wird versucht, sich mit einem drahtlosen Netzwerk zu verbinden. Wenn dieses nicht gelingt oder es konfiguriert ist, wird ein Access Point vom Mikrocontroller erstellt. Mit diesem Vorgehen kann die Uhr trotz mangelnden externen Access Point durch den Benutzer erreicht werden.

Nach der *setup*-Routine wird, bis der Mikrocontroller ausgeschaltet wird, wiederholend die *loop*-Routine ausgeführt. Innerhalb dieser Routine werden vier Schritte vom Programm ausgeführt.

Im ersten Schritt wird eine Referenzzeit berechnet. Diese wird aus der Mikrocontroller-Uhr gewonnen und gibt die Dauer an, wie lange das System angeschaltet ist. Um eine möglichst hohe Präzision zu erzielen werden dabei die Mikrosekunden aus dem Mikrocontroller verwendet. Es musste darauf geachtet werden, dass dieser Wert ca. alle 70 Minuten überläuft.

Aufgrund der unterschiedlichen Betriebsmodi, die unterstützt werden sollen, wurde die Berechnung der Uhrzeit in drei Schritte unterteilt. Die zuvor berechnete Referenzzeit wird zur Synchronisation der Schritte genutzt. Im ersten Schritt wird eine absolute Uhrzeit angefragt. Eine absolute Uhrzeit besteht dabei immer aus einer Uhrzeit und einer Referenzzeit. Die Referenzzeit gibt an, wann die absolute Uhrzeit bestimmt wurde. Als nächstes erfolgt die Bestimmung einer differentiellen Uhrzeit. Diese wird über die Referenzzeit und die derzeitige vorliegende absolute Uhrzeit bestimmt. Nachdem absolute und differentielle Uhrzeit bestimmt sind werden diese addiert und ergeben die anzuzeigende Uhrzeit.

Entsprechend der zuvor berechneten anzuzeigenden Uhrzeit werden im nächsten Schritt die Schrittmotoren angesteuert um die Uhrzeiger zu positionieren. Die Positionierung kann sich dabei über mehrere Zyklen erstrecken. Bei der Positionierung wurde darauf geachtet, dass die Schrittmotoren einfach ersetzt werden können und auch Übersetzungen unterstützt werden, welche nicht zu einer Umdrehung gerade aufgehen (zum Beispiel 3/2048). Das selbe gilt für die Minuten- und Stundenmarkierungen auf dem Ziffernblatt. Damit werden auch Uhren möglich welche zum Beispiel nach drei Umdrehungen auf die Zwölf zeigen sollen. Ähnlich zu den absoluten und differentiellen Zeitgeber kann der Uhrzeitanzeiger im laufenden Betrieb ausgetauscht werden. Auch hier ist die Motivation die Unterstützung von Betriebsmodi.

Im letzten Schritt der *loop*-Routine werden HTTP- und WebSocket-Anfragen bearbeitet. Zum einen erfolgt so die Bereitstellung der Benutzungsschnittstelle und zum anderen die Kommunikation mit dieser. Entsprechend werden in diesem Schritt Konfigurationen angepasst oder Funktionen von Zeitgeber oder Zeitanzeiger aufgerufen.

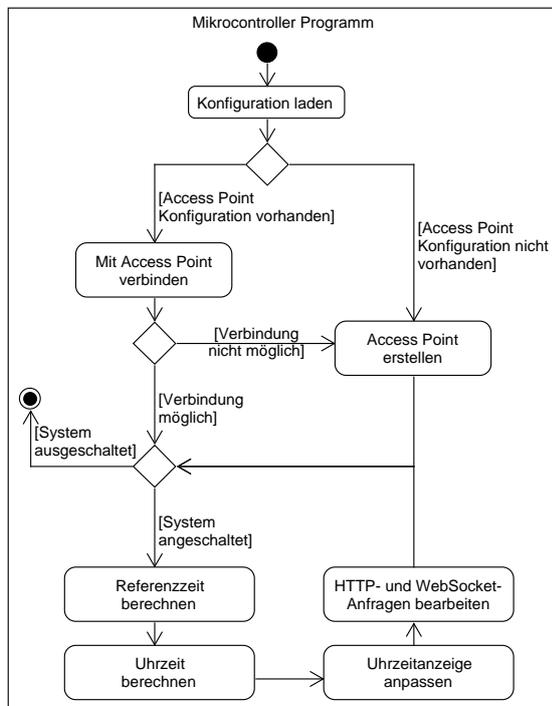


Abbildung 5. Mikrocontroller Programm Aktivitätsdiagramm

5.1. Betriebsmodi

Die Trennung in absolute und differentielle Uhrzeit erlaubt es unterschiedliche Berechnungen miteinander zu kombinieren und so leichter die Betriebsmodi abzubilden. Als absolute Zeitgeber wurden ein *ManuellerZeitgeber*, ein *NTPZeitgeber*, ein *ChaosZeitgeber* und ein *KonstantZeitgeber* umgesetzt. Um die Uhrzeit manuell zu setzen, bietet der *ManuellerZeitgeber* eine Methode an, welche über die Benutzungsschnittstelle aufgerufen werden kann. Eine automatische Uhrzeitstellung wird durch den *NTPZeitgeber* erreicht. Dieser fragt in konfigurierbaren Intervallen die Uhrzeit bei einem NTP-Server an. Zufällige und somit chaotische Uhrzeiten werden vom *ChaosZeitgeber* zu zufälligen Zeitpunkten berechnet. Um die Uhrzeit zum Stillstand zu bewegen, gibt es den *KonstantZeitgeber* welcher einmalig die Uhrzeit festlegt und danach keine Änderung zulässt.

Als differentielle Zeitgeber wurden ein *ReferenzZeitgeber*, ein *ReferenzMultiplikatorZeitgeber* und ein *NullZeitgeber* implementiert. Der *ReferenzZeitgeber* bildet die Differenz zwischen aktueller Referenzzeit und der Referenzzeit, zu der die absolute Uhrzeit gesetzt wurde. Ähnlich verhält sich der *ReferenzMultiplikatorZeitgeber*. Dieser multipliziert die Differenz zusätzlich mit einem konfigurierbaren rationalen Faktor. Um keinen differentiellen Zeitanteil abzudecken, wurde der *NullZeitgeber* implementiert.

Auch wurde der Uhrzeitanzeiger austauschbar gestaltet, damit dieser entsprechend des Betriebsmodus ausgetauscht werden kann. Als Uhrzeitanzeiger wurden ein *ManuellerUhrzeitanzeiger* und ein *AutomatischerUhrzeitanzeiger* implementiert. Der *ManuellerUhrzeitanzeiger* bietet Schnittstellen wodurch der Benutzer die Uhrzeiger manuell verstellen kann. Diese Funktionalität macht eine manuelle physische Kalibrierung der Uhrzeiger überflüssig. Im Regelbetrieb wird der *AutomatischerUhrzeitanzeiger* eingesetzt. Dieser stellt entsprechend der berechneten anzuzeigenden Uhrzeit die Uhrzeiger.

6. Zusammenfassung & Ausblick

In dieser Arbeit wurde das Upcycling einer Kaminuhr inklusive der Analyse, Planung und Realisierung beschrieben. Die Ausgangskaminuhr wurde durch neue Funktionalitäten ergänzt und lästige Eigenschaften, wie die manuelle Umstellung der Zeit, wurden durch den Einsatz eines Mikrocontrollers und entsprechender Software überwunden. Das Upcycling hat die Kaminuhr vor der Entsorgung gerettet. Weiter bietet die Kaminuhr 2.0 wie im Abschnitt 2.2 beschrieben noch viel Upcycling-Potential.

Literatur

- [1] Michael Olsen, <http://www.time-is-fake.de/>, aufgerufen am 14.06.2019
- [2] Microchip, Datenblatt MCP23017/MCP23S17, <http://ww1.microchip.com/downloads/en/devicedoc/20001952c.pdf>
- [3] Espressif Systems, Datenblatt ESP8266, https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [4] Texas Instruments, Datenblatt ULN200x/ULQ200x, <http://www.ti.com/lit/ds/symlink/uln2003a.pdf>
- [5] MikroElektronika, Datenblatt 28BYJ-48, <https://download.mikroe.com/documents/datasheets/step-motor-5v-28byj48-datasheet.pdf>

Kaminuhr 2.0 - Upcycling einer Kaminuhr

Florian Schmalriede

AUSGANGSSYSTEM

- Kaminuhr aus den 1960er Jahren
- defekt und für die Entsorgung vorgesehen
 - ⇒ Modifikationen beliebig möglich
- manuelle Zeiteinstellung notwendig
- manuelles Aufziehen notwendig



UPCYCLING-MÖGLICHKEITEN

- Einbau eines Mikrocontroller-gesteuerten Uhrwerks
 - ⇒ Funktionsumfang kann durch Software ergänzt werden
 - ⇒ manuelles Aufziehen des Uhrwerks entfällt
- webbasierte Benutzungsschnittstelle
 - ⇒ Konfiguration und Steuerung über ein Netzwerk möglich

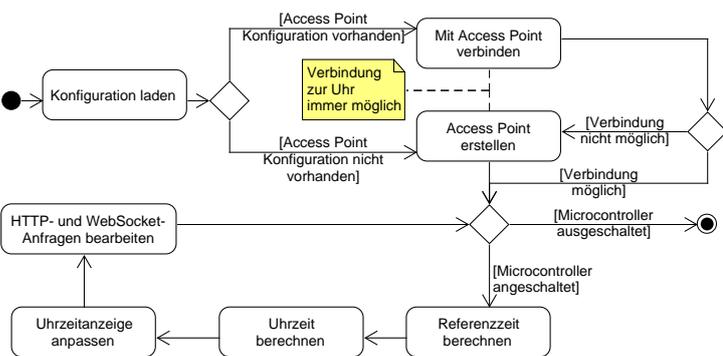
ZIELSETZUNG

- Einbau eines Mikrocontroller-gesteuerten Uhrwerks
 - Steigerung des Funktionsumfangs durch verschiedene Betriebsmodi
- webbasierte Benutzungsschnittstelle
 - über WLAN erreichbar
- getrennte Ansteuerung von Stunden- und Minutenzeiger
 - ⇒ unabhängige Positionierung möglich

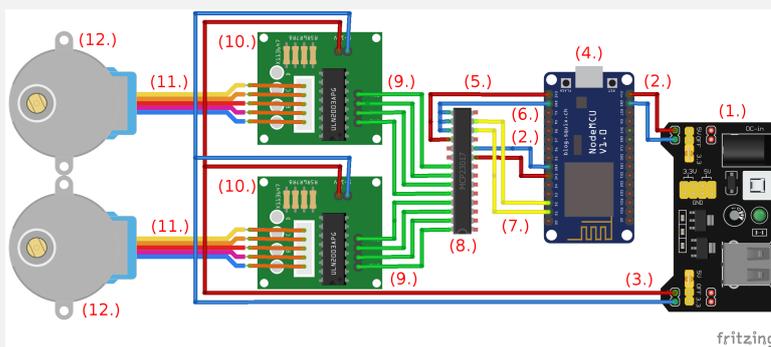
BETRIEBSMODI

- *Manuell*-Betriebsmodus
 - Zeiteinstellung manuell möglich
 - fortlaufende Zeitberechnung durch Referenzzeit
- *NTP*-Betriebsmodus
 - automatische Zeitsynchronisation mit einem NTP-Server
 - ⇒ manuelle Zeiteinstellung entfällt
 - konfigurierbares Synchronisationsintervall
 - fortlaufende Zeitberechnung durch Referenzzeit
- *Kalibrierung*-Betriebsmodus
 - nach Stromverlust sind die Zeigerpositionen nicht bekannt
 - Uhrzeiger können über Benutzungsschnittstelle manuell positioniert werden
- *Zeitreise*-Betriebsmodus
 - Zeiteinstellung manuell möglich
 - fortlaufende Zeitberechnung durch Referenzzeit und konfigurierbaren rationalen Manipulationsfaktor
 - ⇒ die Uhr läuft schneller bzw. langsamer
- *Chaos*-Betriebsmodus
 - zufällige Uhrzeiten zu zufälligen Zeitpunkten bei zufälliger Drehrichtung der Uhrzeiger
 - ⇒ ähnlicher Entspannungseffekt wie bei einer Lavalampe

PROGRAMMABLAUF

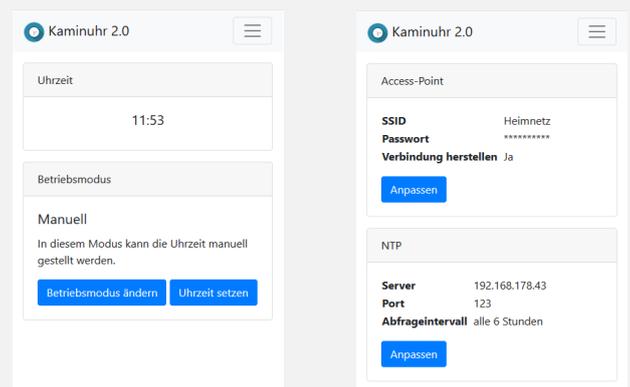


AUFBAU DER HARDWARE



1. Spannungswandler
2. 3,3 V Stromversorgung
3. 5 V Stromversorgung
4. Mikrocontrollerboard NodeMCU mit Mikrocontroller ESP8266
5. IO-Expander aktivieren
6. I2C-Adresse des IO-Expanders
7. I2C-Kommunikation
8. IO-Expander MCP23017
9. digitale Schrittmotorbefehle
10. Schrittmotortreiber ULN2003
11. analoge Schrittmotorbefehle
12. Schrittmotor 28BYJ-48

BENUTZUNGSSCHNITTSTELLE



- Abruf der Benutzungsschnittstelle erfolgt über HTTP
- Kommunikation zwischen Benutzungsschnittstelle und Mikrocontroller erfolgt über WebSocket-Protokoll
 - ⇒ bidirektionale Verbindung
- Beachtung von responsive Webdesign
 - ⇒ Darstellung erfolgt entsprechend des Endgeräts

Erweitertes digitales Barometer zur Erfassung von Schimmel

Niklas Holtz
Informatik (Fach-Bachelor)
Carl von Ossietzky Universität
Oldenburg

Abstract

Das Ziel dieser Ausarbeitung ist es zu prüfen, inwiefern sich ein altmodisches Messing-Barometer zu einer modernen Wetterstation upcyclen lässt. Zusätzlich soll durch eine Erweiterung des Messumfangs der Funktionsumfang so erweitert werden, dass der Bildung von Schimmel vorgebeugt werden kann. Dazu wird zunächst erläutert, was für Schritte nötig sind, um den Funktionsumfang des Barometer entsprechend zu erweitern, wie die Bildung von Schimmel mit Wetterdaten zusammenhängen kann und was für Gefahren von Schimmel ausgehen. Nach erfolgreicher Umsetzung mittels eines Mikrocontrollers in Verbindung mit einem Einplatinencomputer zur Langzeitmessung, konnte in einem abschließenden Fallbeispiel herausgefunden werden, dass Gewohnheiten von Bewohnern maßgeblich zur Bildung von Schimmel beitragen. Morgendliches Lüften und eine permanente Luftzufuhr haben sich als besonders wirksam zur Eindämmung von Schimmel herausgestellt.

1. Einleitung

Kultobjekte wie Messing-Barometer, die durch ihr üppiges Gehäuse direkt ins Auge fallen, werden oft über Generationen weitergereicht und haben eine durchaus ästhetische Optik, die man heute wohl als *retro* bezeichnen würde. Doch nagt an solchen Objekten oft der Zahn der Zeit, da entsprechende Funktionalitäten längst durch die moderne Technik besser umgesetzt werden können.

Mikrocontroller bieten heute glücklicherweise die Möglichkeit solche Kultobjekte zu erhalten und ihren Funktionsumfang zu erweitern oder zu verbessern. Im Rahmen dieser Möglichkeit soll geprüft werden, inwiefern sich ein klassisches Messing-Barometer sinnvoll in seinem Funktionsumfang erweitern lässt. Dazu sollen über den bisherigen Messumfang hinaus Wetterdaten gesammelt werden, die nicht nur Rückschlüsse auf das kommende Wetter geben, sondern auch Anzeichen auf mögliche gesundheitliche Gefahrenquelle im direkten Umfeld aufdecken können.

1.1. Historie des Barometers

Barometer mit Messing-Gehäuse werden gerne als Geschenk zu Hochzeiten oder Geburtstagen überreicht. Durch sie konnten Privathaushalte erstmals bevorstehende Witterungen leicht anhand des Luftdrucks herleiten [1].

Klassischerweise wurden bis 1881 Flüssigkeitsbarometer genutzt, um den Luftdruck zu bestimmen. Quecksilber-Barometer beispielsweise haben den Luftdruck immer aus dem Verhältnis von einer Quecksilbersäule und einer offenen Luftsäule bestimmt. Das sogenannte Aneroid oder Dosenbarometer hingegen misst den Luftdruck anhand des Druckes auf eine luftleere Dose [2]. Das Ausdehnen bzw. Zusammenziehen der Dose wird durch einen Stift auf eine Spiralfeder übertragen, durch die man auf einem unterliegenden Ziffernblatt dann direkt den Luftdruck ablesen kann [3].

Luftdruckmessungen werden heutzutage an Wetterstationen durchgeführt, sodass Privatpersonen keine eigenen Messungen vornehmen brauchen, sondern die Resultate leicht über das Internet beziehen können. Die Barometer, die dort zum Einsatz kommen, sind jedoch winzige elektronische Sensoren, die genauer sind, als ihre mechanischen Vorgänger.

1.2. Die Upcycling-Idee

Natürlich kann man jederzeit genaue Messdaten von Wetterstationen in der Region über das Internet beziehen, doch gibt diese Möglichkeit nicht ohne weiteres für das Eigenheim. Der Luftdruck wird sich im Vergleich zu besagten Wetterstationen kaum unterscheiden, doch können aus anderen Wetterdaten spezifische Informationen über die Wittersituation am Anwendungsort generiert werden. Zudem fällt es jungen Leuten heute zunehmend schwerer analoge Anzeigen, wie die in altmodischen Messing-Barometern, richtig zu deuten. Der erste Schritt der Upcycling-Idee ist es daher, den Messumfang des Barometers durch elektronische Sensoren so zu erweitern, dass Luftfeuchtigkeit, Temperatur und auch der Luftdruck gemessen und durch eine

separate digitale Anzeige leicht einsehbar gemacht werden.

Über den Ist-Zustand der Messungen hinaus können auch zusätzlich Informationen anhand des Verlaufs der Messdaten gewonnen werden. Beispielsweise ist es bei Luftdruckmessungen eher entscheidend, wie schnell der Luftdruck abnimmt, denn je schneller bzw. stärker dieser abfällt, desto heftiger verschlechtert sich das Wetter [3]. Der zweite Schritt der Upcycling-Idee besteht dem zu Folge daraus, eine externe Messdatenspeicherung einzurichten, sodass auch Langzeitmessungen möglich sind, deren Verläufe dann entsprechend analysiert werden können. Diese Verläufe werden schließlich auf einer kleinen Webseite visualisiert, die über das eigene Netzwerk eines externen Gerätes, welches als Datenspeicher fungiert, verfügbar ist.

Ein besonderer Fokus wird auf den Verlauf der Luftfeuchtigkeit und der Temperatur gesetzt. Gesundheitliche Gefahrenquellen, wie beispielsweise Schimmelpilze, werden unter anderem maßgeblich von diesen beeinflusst. Durch die Langzeitmessungen können kritische Messwerte (siehe Kapitel 1.3) konkreten Zeitpunkten zugeordnet werden, sodass der Bildung von Schimmelpilzen effektiv vorgebeugt werden kann.

1.3. Gefährdung durch Schimmel

Sichtbare und auch unsichtbare Schimmelpilze können erhebliche Erkrankungen auslösen. Ihr optimales Wachstum haben sie bei etwa 25 °C erreicht, doch benötigen sie mindestens 0 °C, um sich zu vermehren [4]. Ihr Wachstum wird allerdings maßgeblich von der Luftfeuchtigkeit beeinflusst, da sie sich bei einer Luftfeuchte von über 70% ideal verbreiten können. Jedoch ist bereits ab 60% Luftfeuchtigkeit Schimmelbildung nicht auszuschließen [5].

Die Gesundheitsschäden, die von Schimmel verursacht werden können, sind vielfältig. Einige Schimmelpilze erzeugen Gifte, die dem Menschen schaden können und sogar krebserregend sind. Hauptsächlich nehmen wir die Schimmelpilze jedoch über die Atemwege und den Magen-Darm-Trakt auf, wodurch es von Atemwegsreizungen und Allergien bis hin zu Pilzinfektionen oder Hautausschlägen kommen kann [4]. Erschwerend kommt hinzu, dass diese Symptome schlecht von anderen Erkrankungen zu unterscheiden sind, was die Gesundheitsschäden von Schimmel schwer diagnostizierbar macht.

2. Umsetzung

Zentrales Element der Messstation ist ein Mikrocontroller, der über seine Pins per I²C-Datenbus mit allen weiteren Peripheriegeräten verbunden ist. Der Controller ist zusammen mit einem Sensor auf einer Platine fest verlötet, die wiederum im Inneren des Barometers verklebt ist. Um eine möglichst genaue Messung zu erhalten, sind zusätzlich

Lüftungsschlitze in die Außenfassade des Barometers gebohrt, sodass frische Luft in das Gehäuse gelangt.

Als Stromversorgung ist eine einfache 9V-Blockbatterie über einen Schalter, der in die Außenfassade eingelassen ist, mit dem Controller verbunden. Die Messdaten werden außerdem auf einer Anzeige im Ziffernblatt des Barometers angezeigt. Wie genau der Mikrocontroller geschaltet ist, kann der Abbildung 1 entnommen werden.

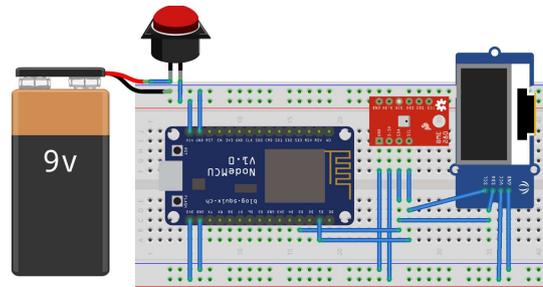


Abbildung 1. Schaltung des Mikrocontrollers.

Als Empfängerstation dient ein Einplatinencomputer, auf dem die Daten im Sinne der Langzeitmessung mittels einer *MYSQL*-Datenbank abgelegt werden sollen. Ergänzend werden mehrere *PHP*-Skripte verwendet, um eine dynamische Webseite zu erzeugen, sodass die Messdaten dort graphisch angezeigt werden können. Eine genaue Auflistung der Bauteile ist der nachfolgenden Tabelle zu entnehmen.

Bauteil	Beschreibung
<i>ESP8266</i>	Mikrocontroller, der die Daten des Sensors über eine WLAN-Verbindung weiterleitet.
<i>BME280</i>	Sensor zur Erfassung von Luftfeuchtigkeit, Temperatur und Luftdruck.
<i>OLED-Display</i>	Anzeige der Messdaten auf dem Barometer (1,3 Zoll, 128x64 Pixel).
Wippschalter	10 A/12 V Wippschalter zum Ein- und Ausschalten des Barometers.
<i>Raspberry Pi 3</i>	Einplatinencomputer zur Speicherung der Messdaten in eine Datenbank mit Ladekabel und Gehäuse.

2.1. Messung und Anzeige

Durch die kompatible *Arduino IDE* [8] erhält der Mikrocontroller ein Skript, durch welches sequentiell bis zu drei Aktionen in einem Schleifendurchlauf ausgeführt werden können. Wie in Code-Auszug 1 zu

sehen ist, wird entweder das Display aktualisiert (7), Daten an den RaspberryPi weitergegeben (14) oder ein Verbindungsaufbau durchgeführt (18), durch den im Falle eines Verbindungsabbruchs erneut eine WiFi-Verbindung aufgebaut werden kann. Die Hilfsmethode *isTimePassed(unsigned long time, unsigned long &ts)* sorgt durch entsprechendes Speichern von Zeitstempeln jeder Aktion für ein genaues Timing der Ausführung.

```

1  /* Zeitstempel */
2  unsigned long lastDataUpload = 0;
3  unsigned long lastDisplayRefresh = 0;
4  unsigned long lastWifiRefresh = 0;
5
6  void loop() {
7      if(isTimePassed(5 * 1000,
8          lastDisplayRefresh))
9          displayData();
10
11     if(WiFi.status() == WL_CONNECTED) {
12         if(isTimePassed(10 * 60 * 1000,
13             lastDataUpload))
14             uploadData();
15     } else {
16         if(isTimePassed(10 * 1000,
17             lastWifiRefresh))
18             refreshConnection();
19     }
20 }

```

Code-Auszug 1. Mikrocontroller-Loop.

2.2. Externer Zugriff

Um den Mikrocontroller mit dem Raspberry Pi zu verbinden, wird letzterer als sog. *WLAN-Access-Point* genutzt, durch den externe Geräte auf das eigene Netzwerk des Raspberry Pis zugreifen können [7]. Der Raspberry Pi ist dadurch im entsprechenden Netzwerk durch die Adresse *192.168.1.1* erreichbar. Durch die *uploadData()*-Methode des Mikrocontrollers wird eine *HTTP-Anfrage* der Messdaten durchgeführt.

```

1  String url = "/data.php?humidity="+
2      String(bme.readHumidity())+
3      "&temperature="+
4      String(bme.readTemperature())+
5      "&air_pressure="+
6      String(bme.readPressure() / 100.0F);
7  client.print(String("GET_") + url +
8      "_HTTP/1.1\r\n" +
9      "Host:_ " + host + "\r\n" +
10     "Connection:_close\r\n\r\n");

```

Code-Auszug 2. Datenübertragung.

Wie in Code-Auszug 2 zu sehen ist, werden dazu die Parameter (2-7) als *GET-Parameter* (8) übergeben, die durch ein PHP-Skript auf der Server-Seite ausgelesen werden (siehe Code-Auszug 3).

```

1  <?php
2      include_once "databasemanager.php";
3      $db = new DatabaseManager();
4      if(isset($_GET["temperature"]) and
5          isset($_GET["humidity"]) and
6          isset($_GET["air_pressure"])) {
7          $success = $db->insertData(
8              $_GET["humidity"],
9              $_GET["temperature"],
10             $_GET["air_pressure"]);

```

Code-Auszug 3. Raspberry Pi.

Beim Einsetzen in die Datenbank (7) wird zudem abhängig von der Server-Zeit ein Zeitstempel generiert, nach dem die Daten sortiert werden. Die Daten werden durch das Open-Source-Framework *Chart.js* [9] graphisch dargestellt. Um die Server-Zeit möglichst einfach von außen anpassen zu können, wurde zusätzlich eine Kommandozeile eingefügt, durch die der Benutzer Befehle ausführen lassen kann.

3. Auswertung

Wie im Quellcode vom Mikrocontroller zu sehen ist, werden die Messdaten in einem Zyklus von 10 Minuten hochgeladen (siehe Zeile 12 in Code-Auszug 1). Dies reicht bereits aus, um auch minimale Schwankungen zu registrieren. Dadurch können nun je nach Bedarf unterschiedliche Informationen generiert werden. Beispielsweise könnten durch den Verlauf der Temperatur Rückschlüsse auf eine gute bzw. schlechte Isolierung des Eigenheims gewonnen werden. Hinsichtlich der gesundheitlichen Gefahren durch Schimmel ist aber vor allem die Luftfeuchtigkeit innerhalb von Räumen ausschlaggebend.

3.1. Fallbeispiel: Schlafzimmer

In einem ersten Versuchsdurchlauf wurde das Barometer in einem Schlafzimmer platziert, welches bereits Anzeichen von Schimmel an den Fensterrahmen aufwies. Da man in diesem Raum regelmäßig mehrere Stunden am Stück verbringt, eignete er sich besonders gut für eine Langzeitmessung.

Um ein möglichst ausgewogenes Messergebnis zu erhalten, wurde das Barometer in die Mitte des Raumes platziert. Abbildung 2 zeigt den Verlauf der Luftfeuchtigkeit über zwei Tage, der durch *Chart.js* auf dem Raspberr Pi visualisiert wurde.

Schwankungen der Gewohnheiten von Bewohnern wurden durch das Verschließen der Tür bzw. durch das Lüften simuliert. In der ersten Nacht (21.5.) wurde dazu die Tür geschlossen gelassen und am Morgen ausreichend gelüftet. In der zweiten Nacht (22.5.) wurde dagegen mit offener Tür, jedoch ohne anschließendem Lüften gemessen.



Abbildung 2. Luftfeuchtigkeit über zwei Tage.

Wie zu sehen ist, steigt die Luftfeuchtigkeit in beiden Fällen auf ein kritisches Niveau (d.h. über 60% über einen längeren Zeitraum). In der ersten Nacht sinkt die Luftfeuchtigkeit sehr stark am Morgen, was auf das Lüften zurückzuführen ist. Danach wird also keine Bildung von Schimmel mehr gefördert.

Die Kurve der zweiten Nacht erreicht nicht so hohe Messwerte wie die erste, jedoch sinkt die Kurve viel langsamer ab. Das Fehlen des Lüftens macht sich hier direkt bemerkbar. Wäre in der Nacht ein noch höherer Wert erreicht worden, hätte man die Bildung von Schimmel möglicherweise noch weit in den Tag hinein gefördert.

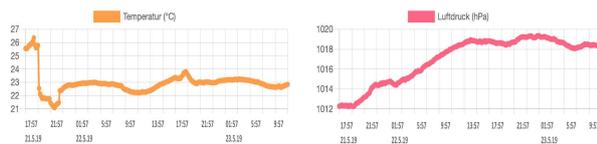


Abbildung 3. Luftdruck und Temperatur über zwei Tage.

Wie in Abbildung 3 zu sehen ist, gibt es zwar geringe Schwankungen der Temperatur und einen Anstieg des Luftdrucks, doch lassen sich daraus keine direkten Erkenntnisse gewinnen, da die Änderungen zu gering sind. Jedoch kann man sagen, dass das Wachstum von Schimmelpilzen in diesem Temperaturbereich beinahe optimal begünstigt wird, da die 25 °C-Marke, bei der Schimmelpilze ein optimales Wachstum aufweisen [4], beinahe erreicht wurde.

3.2. Ausblick

Über die Prävention von Schimmel hinaus, lässt sich das modernisierte Messing-Barometer natürlich auch einfach als Wetterstation nutzen. Um diesen Zweck optimal zu erfüllen, wären noch weitere Verbesserungen des Funktionsumfangs vorstellbar. Um noch genauere Messdaten zu erhalten, könnte man einen weiteren Sensor an die Außenfassade des Barometers befestigen, der durch den direkteren Kontakt zu frischer Luft noch genauere Messungen vornehmen würde. Außerdem könnte die sehr kleine digitale Anzeige durch eine größere ersetzt werden. Hinsichtlich der kritischen Luftfeuchtigkeit in Bezug auf Schimmel könnte eine extra LED angebracht werden, die gegebenenfalls blinkt, falls der Wert (60%) überschritten wurde.

4. Fazit

Das Barometer vermittelt nun nicht nur interessante Wetterinformationen, sondern hat sich auch als ein nützliches Instrument zur Vermeidung von Schimmel herausgestellt. Zudem ist es durch seine moderne Technik (insbesondere der hervorstechenden Anzeige) und sein altmodisches Design ein Spagat aus neu und alt. Durch die digitale Anzeige kann nun auf einen Blick der Ist-Zustand des Luftdrucks, der Luftfeuchtigkeit und der Temperatur abgelesen werden und aus dem Verlauf der Messwerte auf dem Raspberry Pi können je nach Nutzung viele neue Informationen generiert werden.

Die Langzeitmessung in einem Schlafzimmer hat gezeigt, dass gerade Räume, in denen man mehrere Stunden verbringt, ein besonderes Augenmerk im Bezug auf die Bildung von Schimmel benötigen. Die beiden Nächte haben gezeigt, dass gutes Lüften am Morgen die Bildung von Schimmel direkt unterbindet. Außerdem kann eine permanente Luftzufuhr durch eine geöffnete Tür ebenfalls die Bildung von Schimmel eindämmen.

Die gemessenen Daten des Luftdrucks und der Temperatur haben sich dabei kaum als hilfreich erwiesen, da sich die Veränderungen der Werte auf ein Minimum beschränkt haben. Es kann jedoch festgestellt werden, ob die gemessene Temperatur die Bildung von Schimmel begünstigt. Es ist nicht auszuschließen, dass die Datenspeicherung der Werte an anderen Anwendungsorten und zu anderen Zwecken, wie die Prüfung auf eine ausreichende Isolierung, sehr viel aufschlussreicher sein kann.

Es hat sich außerdem gezeigt, dass das modernisierte Messing-Barometer weiterhin viel Potential zum Upcycling aufweist. Je nach Anwender, Anwendungsort und Zweck könnte man es immer weiter verbessern und trotzdem stets den Charme der ursprünglichen mechanischen Messtechnik erhalten.

Literatur

- [1] Joseph-Willhelm Fischer, *Das Barometer als Witterungsvoranzeiger*, Anton Strauß Verlag, 1827.
- [2] Theodor Nowak, *Das barometrische Höhenmessen mit dem Aneroid (Dosenbarometer)*, Lehmann und Wentzel, 1869.
- [3] Edmund Herzog, *Praktische Anleitung zum Höhenmessen mittelst Dosenbarometer*, Wilh. Engelmann Verlag, 1873.
- [4] Michael Köneke, *Schimmel im Haus: erkennen - vermeiden - bekämpfen*, ISBN: 9783816764243, Frauenhofer IRB Verlag, 2008.
- [5] Eva Lübbe, *Feuchteschutz*, ISBN: 9783322912015, GWV Fachverlage GmbH, Wiesbaden 2007.
- [6] Francesco Azzola, *Android Things Projects*, ISBN: 9781787289246, Packt Publishing LTD, 2017.
- [7] Christian Immler, *Raspberry Pi: Mach's einfach: Die kompakteste Gebrauchsanweisung*, ISBN: 9783645603515, Franzis Verlag GmbH, 2018.
- [8] Arduino IDE, <https://www.arduino.cc/en/Main/Software>, abgerufen am 27.06.2019.
- [9] Chart.js - Open Source JavaScript Framework, <https://www.chartjs.org/>, abgerufen am 27.6.2019.

Erweitertes digitales Barometer zur Erfassung von Schimmel

Niklas Holtz

DIE UPCYCLING-IDEE

- modernisierte Wetterstation für das Eigenheim
 - neuer Sensor zur Messung des Luftdrucks
 - Erweiterung des Messumfangs, um auch Luftfeuchtigkeit und Temperatur zu messen
 - digitale Anzeige der Wetterdaten
 - erweiterbar in der Funktionalität
 - Erhaltung des Retro-Charmes
- Möglichkeit der Langzeitmessung
 - externer Raspberry Pi als Speichermedium
 - drahtlose Verbindung mittels WiFi
 - Visualisierung der Daten durch Graphen auf einer Webseite
 - Prävention von Schimmel im Eigenheim durch Analyse der Daten und Verhaltensänderung



GEFÄHRDUNG DURCH SCHIMMEL

- Bildung benötigt 0 °C bis 25 °C und mindestens 60% Luftfeuchtigkeit
- vielfältige Gesundheitsschäden:
 - manchmal krebserregend
 - Atemwegsreizungen
 - Allergien
 - Pilzinfektionen
 - Hautausschläge
- schwer diagnostizierbar

DATENÜBERTRAGUNG

```
String url = "/data.php?humidity="+
String(bme.readHumidity())+
"&temperature="+
String(bme.readTemperature())+
"&air_pressure="+
String(bme.readPressure() / 100.0F);
client.print(String("GET ") + url +
" HTTP/1.1\r\n" +
"Host: " + host + "\r\n" +
"Connection: close\r\n\r\n");
```

Code-Auszug 1: Sender.

- HTTP-Anfrage der Messdaten
- Werte als GET-Parameter (Zeile 7)

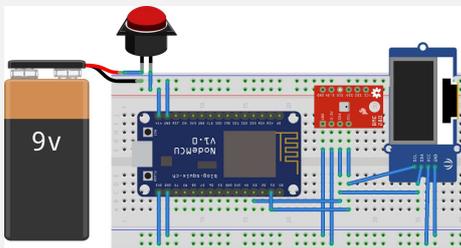
```
<?php
include_once "databasemanager.php";
$db = new DatabaseManager();
if (isset($_GET["temperature"]) and
isset($_GET["humidity"]) and
isset($_GET["air_pressure"])) {
    $success = $db->insertData(
        $_GET["humidity"],
        $_GET["temperature"],
        $_GET["air_pressure"]);
```

Code-Auszug 2: Empfänger.

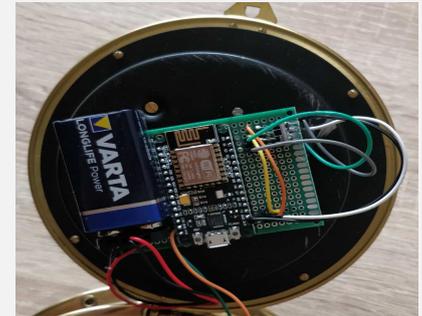
- Einsetzen in die Datenbank
- Generierung eines Zeitstempels

UMSETZUNG UND HARDWARE

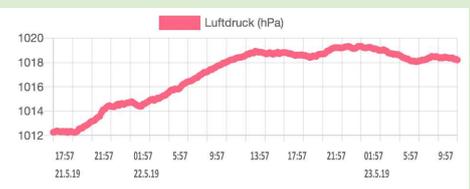
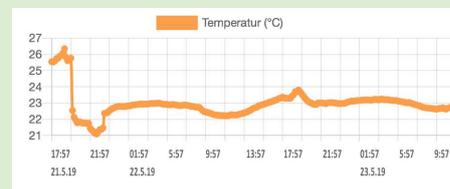
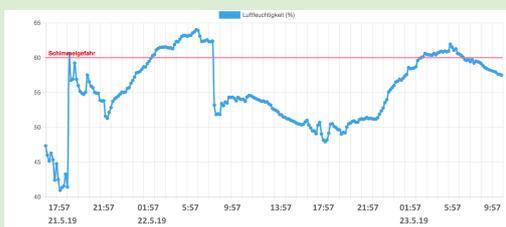
- Schaltung des Microcontrollers



- Microcontroller (ESP8266) dient als zentrales Element und Sender der Messdaten
- Sensor zur Erfassung von Luftfeuchtigkeit, Temperatur und Luftdruck (BME280)
- OLED-Display zur Anzeige der Messdaten (1,3 Zoll)
- Wippschalter zum Ein- und Ausschalten
- 9v-Blockbatterie als Stromversorgung
- Raspberry Pi 3 als externer Empfänger der Messdaten



FALLBEISPIEL: SCHLAFZIMMER



- Simulation von Gewohnheiten der Bewohner:
 - Nacht des 21.5.: Verschließen der Tür über Nacht
 - Nacht des 22.5.: Ausreichendes Lüften am Morgen
- Luftfeuchtigkeit steigt in beiden Fällen auf über 60%

- Temperatur begünstigt die Bildung Schimmel beinahe optimal
- Schwankungen von Luftdruck und Temperatur zu gering für direkte Auswirkungen
- Lüften senkt die Luftfeuchtigkeit (siehe 5:57 Uhr am 22.5) schlagartig
- offene Tür (siehe die Nacht zum 23.5) verringert den Anstieg des Luftfeuchtigkeit

Fazit

- beide Maßnahmen eignen sich gut zur Prävention von Schimmel
- großes Potenzial auch für andere Anwendungsorte (z.B. Rohbauten, Keller)

Upcycling mit Raspberry Pi und Co.: Internetradio

Mykhailo Ievreinov

Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften
Carl von Ossietzki Universität
Oldenburg

Abstract

Ziel des Proseminars "Aus alt mach neu: Upcycling mit Raspberry Pi & Co." war es aus einem alltäglichen, heute nicht mehr verwendeten Gegenstand unter Verwendung eines Einplatinencomputers wie dem Raspberry Pi einen modernen zeitgemäßen Gebrauchsgegenstand herzustellen. Für das Projekt wurde ein ausgedientes Boxensystem für Computer ausgewählt. Aus diesem wurde ein modernes, WLAN-fähiges Internetradio mit Touchscreen und LAN- sowie USB-Anschluss gebastelt. Es wird das Potential von Raspberry Pi im Kontext mit dem Upcycling des Boxensystems untersucht. Der Prozess der Entwicklung des Internetradios wird unter Verwendung des Betriebssystems Raspbian beschrieben. Hierbei interessiert die Frage: Lässt sich das Internetradio unter Verwendung dieser Software steuern?

1. Einleitung

Die Entwicklung der modernen Technik schreitet immer schneller voran. Dies führt dazu, dass die Lebensdauer der Geräte immer kürzer wird.

Mehr als die Hälfte derjenigen, die ihr Handy wechseln, tun dies nicht, weil das alte Mobiltelefon kaputt ist. Der häufigste Grund für einen Neukauf ist der Wunsch nach einem noch besseren Gerät. Laut Stiftung Warentest tauschen 42 Prozent aller Mobiltelefon-Nutzer in Deutschland ihr Gerät innerhalb von zwei Jahren aus. 16 Prozent nutzen das Handy drei Jahre lang. Weitere 12 Prozent besorgen sich im Schnitt alle vier Jahre ein neues Handy. Nur etwa 20 Prozent der Handy-Besitzer telefonieren mit ihrem Gerät fünf Jahre oder länger[1].

Dies führt zu immer mehr Elektroschrott und Geräten, die zwar noch funktionsfähig sind, aber nicht mehr dem aktuellen Stand der Technik entsprechen. Upcycling ermöglicht es die Lebensdauer zu verlängern, oder einem ausgedienten Gerät ein neues Leben einzuhauchen. Upcycling ist ein Wortspiel aus den englischen Wörtern „Up“

(nach oben) und „Recycling“ (Wiederverwertung). Anders als beim Recycling wird durch Upcycling ein Mehrwert geschaffen, die stoffliche Aufwertung steht auch im Vordergrund. Somit hat im Idealfall die entstandene Kreation einen höheren Wert als zuvor[2]. Einer der Gründe für Upcycling liegt auf der Ressourcenschonung. Der zweite Grund dafür, alten Dingen ein neues Leben einzuhauchen ist, dass sich Menschen neue Gegenstände finanziell oft nicht leisten können.

In den letzten Jahren hat der Raspberry Pi den Markt immer weiter erobert und ist zu einem beliebten Mini-computer, sowohl für Anfänger, als auch für Fortgeschrittene geworden[3]. Besonders seine vielfältigen Anwendungsmöglichkeiten machen ihn sehr interessant für das Upcycling. Zum Beispiel ermöglicht Raspberry Pi den Einsatz des zur entwickelnden Internetradios.

Die Schwerpunkte dieser Proseminararbeit sind:

1. Die Anbindung von Raspberry Pi an ein altes Boxensystem.
2. Das Hören von Radiosendern über das Internet durch die Software gestützte Ansteuerung des Systems.
3. Die Erhaltung und Erweiterung der alten Funktionalitäten des Geräts.

Das Thema ist interessant, da immer mehr Menschen dazu übergehen, individuelle, Internet gestützte Radiosender zu hören und dabei die Vorzüge moderner Technik wie z.B. einen Touchscreen zu nutzen.

2. Upcycling Möglichkeiten

2.1. Basissystem

Das bestehende Boxensystem aus der Abbildung 1, das upgecyclet werden soll, besteht aus einem Subwoofer mit 10 Watt Leistung und zwei Lautsprecher mit jeweils 3 Watt. Am Frontpanel des Subwoofers befinden sich Bass- und Lautstärkereger. Auf der Rückseite steht ein 3,5 mm



Abbildung 1. Altes Boxensystem zum Upcyceln.

Stereo-Anschlusskabel zur Verfügung. Über dieses Kabel können verschiedene Endgeräte wie Computer, Handy oder MP3-Player angeschlossen werden und damit Musik abzuspielen. Nach dem Upcycling des Systems wird es möglich sein, die Radiosender über das Internet hören zu können. Außerdem bekommt es einen Touchscreen, worüber sich die Lautstärke und Radiosender steuern lassen wird. Auf dem Display werden schließlich die folgende Informationen angezeigt:

- Das Datum und die sekundengenaue Uhrzeit.
- Der Name der Radiostation und des Interpreten.
- Die Bitrate in Bit pro Sekunde.
- Die Lautstärke.

Außerdem wird es möglich sein, die Liste der verfügbaren Radiosender ohne Umschalten anzuschauen.

2.2. Technisches Zubehör

Es sind einige Dinge erforderlich, um ein eigenes Internetradio aus alten Boxensystem bauen zu können.

Hauptsächlich wird ein Raspberry-Pi 2 als Computer-Controller verwendet. Er hat genug Leistung für die oben genannten Ziele des Projekts, eine 900-MHz-Quad-Core-ARM-Cortex-A7-CPU und 1 GB Arbeitsspeicher. Der Computer ist kompakt - passt gut hinter das Display in dem kleinen Gehäuse des Boxensystems. Weiterhin ist relativ einfach, ihn zu programmieren.

Das Internetradio wird über den 3,5 Zoll Touchscreen angesteuert. Zum Anschließen des Touchscreens werden 40 GPIO-Pins benutzt.

Ein Raspberry Pi 2 könnte lediglich über ein LAN-Kabel mit dem Internet verbunden werden. Dies ist aber nicht immer technisch möglich und wenig benutzerfreundlich. Deswegen ist ein WIFI-Adapter nötig, um eine WLAN-Verbindung zu herzustellen.

Für die Eingabe von Befehlen muss noch die PC-Tastatur angeschlossen werden.

2.3. Anwendungsbeispiel

Die rasante Entwicklung des Internetradios aufgrund der breiten Möglichkeiten dieser Technologie verbessert qualitativ die Dienstleistungen, die den Zuhörern zur Verfügung gestellt werden. Im gewöhnlichen Radio kommen Radiowellen durch die Luft. Wenn man in einer kleinen Stadt oder auf dem Land lebt, wird die Auswahl der Radiosender sehr beschränkt. Aber Online-Radio sendet über Internet, deswegen bekommt man eine große Auswahl an Internet-Radiosendern aller Richtungen und Themenbereiche. [4].

Internet-Radio hat die folgenden Hauptvorteile:

- Globale Reichweite. Die Sendezone ist nicht beschränkt, wie bei der Übertragung über den "Äther" - die Reichweite des Repeater. Der Zugriff auf ein bestimmten Radiosender ist überall auf der Welt möglich, solange er mit dem Internet verbunden ist.
- Hohe Qualität des Signals. Die verwendeten Software-Tools ermöglichen es, ein verlustfreies Signal zu liefern ohne Rauschen oder Störfrequenzen.
- Multimedia-Begleitung. Die Ausstrahlung eines Audiosignals (beispielsweise einer Musikkomposition) kann von der Übertragung alphanumerischer oder grafischer Informationen begleitet sein, wie z.B. Titel, Interpret, Hintergrundinformationen.

3. Architektur

3.1. Aufbau der Hardware

Wie auf der Abbildung 2 gezeigt wird, ist im Subwoofer-Gehäuse wurde ein Touchscreen-Display eingebaut. Damit Touchscreen keine extra Stromversorgung braucht, befindet sich gleich dahinter die Raspberry Pi 2, an die das Display über GPIO-Pins angeschlossen ist.

Über Raspberry Pi 2 werden die Internetverbindung hergestellt und die Benutzerbefehle wie Änderung des und der Lautstärke umgewandelt. Um die alten Funktionalität des Geräts zu erhalten, wird zwischen dem kombinierten 3,5-mm-Audioanschluss des Mikrocomputers und dem Microchip des Boxensystem einen Audio-Adapter eingebaut. Dieser bekommt das Signal entweder von Raspberry Pi 2 oder

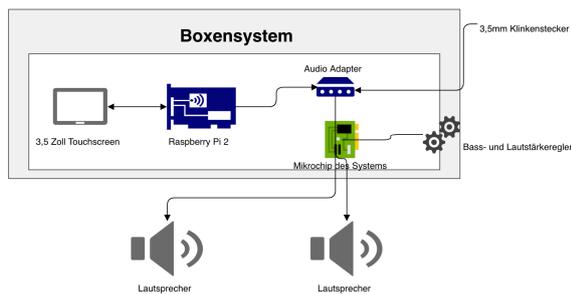


Abbildung 2. Hardware-Architektur des Internetradios

von externen Audioquellen wie z.B. einem PC oder einem MP3-Player und übergibt das Signal an die Lautsprecher. An den Mikrochip ist wiederum der Bass- und Lautstärke regler angeschlossen.

Für eine effizientere Kühlung der CPU und des Video-Chipsatzes werden zusätzliche Kühlkörper eingebaut. Für die Installation des Betriebssystems Raspbian ist eine Speicherkarte mit mindestens 4 GB RAM benötigt. Für den korrekte und schnelle Arbeit des Betriebssystems muss die Geschwindigkeitsklasse der Speicherkarte mindestens 4 sein.

3.2. Software

Ein Computer ohne Betriebssystem ist nutzlose Technik. Weil das Betriebssystem Raspbian zunächst nur Treiber für HDMI-Bildschirme enthält, müssen als erstes die Treiber für den Touchscreen installiert werden. Dafür muss man eine Verbindung zum HDMI-Bildschirm und zum Internet hergestellt werden. Wenn die Touchscreen-Software installiert ist, kann das Display direkt verwendet werden.

Für die Verarbeitung von Audio-Streams wird eine entsprechende Bibliothek wie FFmpeg verwendet. Diese Open-Source-Bibliothek codiert und dekodiert Multimedia-Streams. FFmpeg ist ein hyperschneller Echtzeit Audio/Video Encoder und ein generischer Audio- und Video-Datei-Konverter.

Als Musik-Player, der Musik aus dem angegebenen Verzeichnis spielt wird der Music Player Daemon(MPD) installiert. Der Music Player Daemon selbst ist eine freie und quelloffene Software. MPD wurde für die Client-Server-Architektur entwickelt, wo die Clients über das Netzwerk mit MPD interagieren. MPD ist serverseitige Anwendung zum Abspielen von Musik. Um den MPD zu verwalten, wird der MPD-Client zusammen mit der Bibliothek mpd-Python installiert.

Die Raspberry Pi Internetradio Software ist als Debian-Paket verteilt. Nach der Herunterladung des Pakets wird das Radio Daemon installiert. Danach wird es für den korrekte Arbeit in Übereinstimmung mit Hardware und nach dem gewünschten Design konfiguriert.

4. Zusammenfassung

Upcycling ist eine Form des Recyclings bei der das Erzeugnis daraus einen höheren Wert hat. Einer der Hauptgründe für Upcycling ist dabei das Einsparen an Rohstoffen. Durch Upcycling können Rohstoffe geschont werden, in erster Linie steht aber das Schaffen eines kreativen und individuellen Objektes im Vordergrund.

Die Kosten für die Einrichtung eines Internetradios sind im Vergleich zum Kauf eines neuen Systems relativ gering. Dadurch können nicht nur die Ressourcen geschont ,sondern auch Geld gespart werden.

Die technologische Basis des Internetradio besteht aus drei Elementen:

Station - erzeugt einen Audiostream (entweder aus einer Liste von Audiodateien oder durch direkte Digitalisierung von einer Audiotarte oder durch Kopieren eines vorhandenen Netzwerkstreams) und sendet ihn an den Server.

Server - empfängt einen Audio-Stream von der Station und leitet seine Kopien an alle mit dem Server verbundenen Clients weiter.

Client - empfängt den Audiostream vom Server und wandelt ihn in ein Audiosignal um, das der Hörer des Internetradiosenders hört. Als Client kann jeder Media Player verwendet werden, der Audio-Streaming unterstützt und das Audioformat, in dem das Radio sendet, dekodieren kann.

Neben dem Stream der Audiodaten werden in der Regel auch Textmetadaten übertragen, sodass der Player die Informationen zum Sender und zum aktuellen Titel anzeigt.

Literatur

- [1] Proplanta, URL: https://www.proplanta.de/Agrar-Nachrichten/Umwelt/Lebensdauer-vieler-Geraete-wird-immer-kuerzer_article1450939095.html , 2015
- [2] Florian Hermüller, "Was ist Upcycling?", URL: <https://bean-spectacles.com/blog/2018/02/08/was-ist-upcycling/>, Beans, 2018
- [3] "What is a Raspberry Pi ?" URL: <https://www.raspberrypi.org/documentation/faqs/>
- [4] Michael Schmitz, Wolf Siebel, "Sender & Frequenzen 2012 – Jahrbuch für weltweiten Rundfunkempfang", Baden-Baden 2012, ISBN 978-3-88180-865-1

Upcycling mit Raspberry Pi und Co.: Internetradio

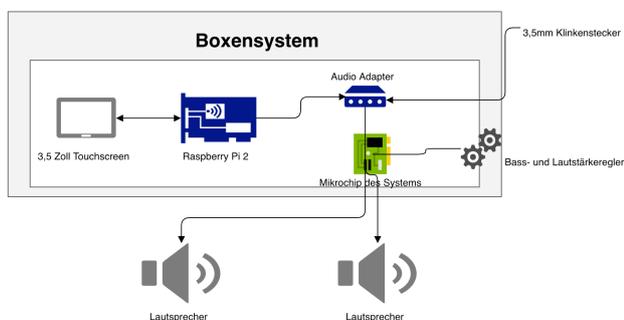
Mykhailo Ievreinov

MOTIVATION

- Systemfähigkeiten des alten Boxensystems:
 - Ein 3,5 mm Stereo-Anschlusskabel.
 - Über dieses Kabel können verschiedene Endgeräte wie Computer, Handy oder MP3-Player angeschlossen werden und damit Musik abzuspielen.
- Nach dem Upcycling wird es möglich sein:
 - die Radiosender über das Internet hören zu können.
 - über einen 3,5 Zoll Touchscreen die Lautstärke und Radiosender steuern.
 - die Informationen zum Datum und zur Uhrzeit, zur Lautstärke, sowie zum Name der Radiostation und des Interpreten über Display anzuschauen.

FUNKTIONALITÄT UND TECHNISCHE ASPEKTEN

- Das Display ist über GPIO-Pins an die Raspberry Pi 2 angeschlossen.
- Über Raspberry Pi 2 werden die Internetverbindung hergestellt und werden die Audiostreams verwaltet.



- Es gibt zwei Möglichkeiten, die Lautstärke einzustellen:
 1. Mit dem Touchscreen.
 2. Mit dem Lautstärkeregler am Frontpanel des Subwoofers.

- Die technologische Basis des Internetradio besteht aus drei Elementen:
 - Station - erzeugt einen Audiostream und sendet ihn an den Server.
 - Server - empfängt einen Audio-Stream von der Station und leitet eine Kopie an mit dem Server verbundenen Client weiter.
 - Client - empfängt den Audiostream vom Server und wandelt ihn in ein Audiosignal um.
- Eine Open-Source-Bibliothek FFmpeg dekodiert die Multimedia-Streams.
- Der Music Player Daemon(MPD) ist serverseitige Anwendung zum Abspielen der Musik.
- Hauptsächlich wird der MPC als Client verwendet.
- In das Verzeichnis /var/lib/mpd/playlists kommt die Playlist mit den einzelnen Radiostreams. Der Dateiname muss mit .m3u enden.
- Der Inhalt der Playlist-Datei enthält eine Liste der Radiostreams im folgenden Format:

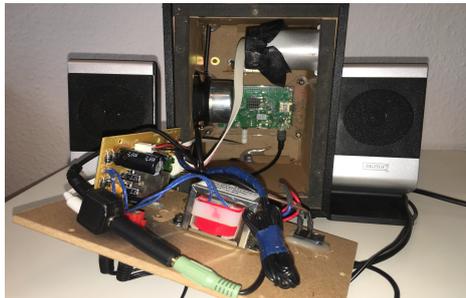

```
#EXTM3U
#EXTINF: -1, [Name des Radiosenders]
[URL für Radio-Stream]
```

UPCYCLING-PROZESS

• Altes Boxensystem vor dem Upcyclen



• Bild der Elektronik



• Abspielen des Internetradios



TECHNISCHES ZUBEHÖR UND KOSTEN

- Raspberry Pi 2 - 33 Euro
 - Netzteil 5V 2.5A und Micro-USB Kabel - 10 Euro
 - Micro-SD Speicherkarte - 5 Euro
 - Audiokabel mit Adapter - 5 Euro
- 3,5 Zoll Touchscreen mit Gehäuse - 25 Euro
- Altes Boxensystem
 - ein Subwoofer mit 10 Watt Leistung
 - zwei Lautsprecher mit jeweils 3 Watt Leistung

Upcycling eines ferngesteuerten Auto

Paula Bürmann

Systemsoftware und verteilte Systeme
Carl von Ossietzky Universität Oldenburg

Abstract

Beim "Upcycling" werden Stoffe und Gegenstände wieder aufgewertet. In diesem Beispiel wird der Prozess des "Upcycling" an einem ferngesteuerten Auto gezeigt. Mithilfe eines Raspberry Pi und einer Kamera wird der Nutzen des Autos erweitert. Das Auto findet wieder eine Verwendung. Dabei werden verschiedene Möglichkeiten zum "Upcycling" analysiert.

1. Einleitung

Der Begriff "Upcycling" beschreibt eine Wiederverwertung von Stoffen und Gegenständen, welche sonst keine Verwendung mehr finden würden. Diese werden durch das "Upcycling" wieder aufgewertet. Dies dient einer nachhaltigen Entwicklung und einer Reduzierung des Material- und Energieverbrauchs. [8] Im Folgenden wird der Prozess des "Upcycling" an einem ferngesteuerten Auto durchgeführt. Zum einen wird die Entwicklung von ferngesteuerten Autos beschrieben und zum anderen welche Möglichkeiten des "Upcycling" vorhanden sind. An einem möglichen Beispiel, wie sich das ferngesteuerte Auto wieder aufwerten lässt, wird die Hardware- sowie Softwarearchitektur dargestellt. Abschließend wird definiert, ob der Upcycling Prozess erfolgreich war und das Auto aufgewertet wurde.

1.1 Entwicklung von ferngesteuerten Autos

Die Entwicklung von ferngesteuerten Autos lässt sich mit der Geschichte der Funktechnik gleichstellen. Durch die Möglichkeit zur Funktechnik werden ebenfalls die ferngesteuerten Autos entwickelt.[3] Die Radiowellen werden erstmalig von dem schottischen Physiker James Clerk Maxwell 1884 erkannt und bereits zwei Jahre später von dem deutschen Forscher Heinrich Herzt in der Praxis genutzt.[6] 1985 wird die erste Funkübertragung durch einen Knallfunksender entwickelt. Dieses ermöglicht zum einen die Übertragung von gesprochenen Nachrichten und

zum anderen die Funksteuerung. Im Jahr 1960 werden die ersten ferngesteuerten Autos mit einem Verbrennungsmotor vorgestellt. Der Markt um ferngesteuerte Autos wächst und die ersten Autos mit Elektroantrieb kommen auf den Markt. Die Austragung der ersten Weltmeisterschaft in den 1980er-Jahren treibt die Forschung an und die Autos erreichen eine Geschwindigkeit von 100 km/h. Der Markt teilt sich in zwei Bereiche und es werden hochwertige sowie kostenintensive Produkte entwickelt als auch günstige Modelle, welche zum Spielen verwendet werden. [3]

2. Durchführung des Upcycling

2.1 Möglichkeiten zum Upcycling

Ferngesteuerte Autos werden über eine Funkfernbedienung mit einer bestimmten Reichweite gesteuert. Dadurch ist die Art der Verwendung begrenzt. Der Benutzer muss sich im selben Raum befinden wie das Auto. Die Verwendung des ferngesteuerten Autos ist eingeschränkt und könnte dazu führen, dass das Auto nach einiger Zeit nicht mehr benutzt wird. Mithilfe von "Upcycling" lässt sich die Reichweite ausbauen und die Verwendungsmöglichkeiten erweitern. Zum "Upcycling" wird ein Minirechner eingebaut, wobei es sich um einen Arduino oder Raspberry Pi handeln kann.

Eine Möglichkeit beinhaltet das Einbauen von Sensoren, welche beispielsweise eine Wand erkennen und vor dieser anhalten. Es können Ultraschallsensoren oder Bewegungssensoren verwendet werden. Das Auto bremst selbständig und benötigt keinen Einfluss durch einen Benutzer. Dadurch besteht ebenfalls die Möglichkeit die Autos autonom fahren zu lassen. Mithilfe des Minirechners kann das ferngesteuerte Auto über eine App vom Handy gesteuert werden oder über ein Webinterface. Durch eine Kamera hat der Benutzer die Möglichkeit zu sehen, wo das Auto lang fährt und kann sich somit in einem anderen Raum befinden, um das Auto zu steuern.

2.2 Realisierung des Upcycling

Im Folgenden wird die Umsetzung des Upcycling“ beschrieben. Die Beschreibung der Umsetzung wird in die Bereiche Hardwarearchitektur und Softwarearchitektur unterteilt. Das ferngesteuerte Auto wird mithilfe eines Raspberry Pi Zero W sowie einer Kamera erweitert. Somit lässt sich das Auto über ein Webinterface steuern und zeigt einen Live-Stream. Der Nutzer kann das Auto von einem anderen Standort steuern und ist nicht mehr an die Reichweite der Funkfernbedienung gebunden. Dieses erweitert die Nutzungsmöglichkeiten im Gelände mithilfe der verbesserten Reichweite.

2.2.1 Hardwarearchitektur

Das ferngesteuerte Auto wird betriebsbereit geliefert. Es handelt sich um das RC Red Thunder der Marke Dickie Toys. Das ferngesteuerte Auto ist ebenfalls für die Außenumgebung geeignet durch die witterungsbeständige ABS-Karosserie. Dabei kann das Auto eine Geschwindigkeit von bis zu 8 km/h aufnehmen und wird durch die Stoßstange geschützt. [2]

In dem ferngesteuerten Auto sind zwei Schrittmotoren sowie eine Schrittmotorsteuerung verbaut. Mit einer 2-Kanal und 40 MHz Funkfernsteuerung wird das Auto gesteuert und kann vorwärts, rückwärts sowie nach rechts und links fahren. [2] Für das "Upcycling" des RC Red Thunder werden folgende Hardwarekomponenten benötigt: Eine L298N DC Schrittmotorsteuerung, welche die Schrittmotoren betreibt. Die Schrittmotoren sind bereits im Auto eingebaut. Des Weiteren wird ein Raspberry Pi Zero W verwendet. Auf diesem sind die notwendigen Softwarekomponenten installiert, damit das RC Red Thunder über das Webinterface gesteuert werden kann. Für die Stromversorgung ist ein Strom PI an dem Raspberry Pi Zero eingebaut, welcher die Hardware durch AA Batterien mit Strom versorgt. Als Kameramodul wird ein Raspberry Pi Kameramodul verwendet.

Die Grundlagen der Auswahl der Hardwarekomponenten orientiert sich dabei an bereits erstellte Projekte mit einem ferngesteuerten Auto und einem Raspberry Pi. Somit konnte die notwendige Stromspannung definiert werden. Des Weiteren wird die L289N Schrittmotorsteuerung eingebaut, welche ebenfalls bei einigen Projekten verwendet wird.[1] Für dieses Projekt wird ein Raspberry Pi Zero W verwendet. Dieser benötigt weniger Strom als beispielsweise der Raspberry Pi 2, ist kleiner und reicht für diesen Anwendungsfall aus. Mithilfe einer Anleitung wird die Schrittmotorsteuerung mit den Schrittmotoren und dem Raspberry Pi W verbunden. Mit diesem Schrittmotor können bis zu zwei Gleichstrommotoren betrieben werden

mit einer Antriebsspannung von 5V-35V. [5]

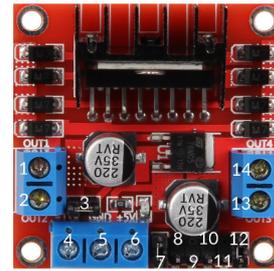


Figure 1: L298N Schrittmotorsteuerung [5]

Die folgende Abbildung zeigt die Verbindung zwischen der Schrittmotorsteuerung, den Schrittmotoren sowie dem Raspberry Pi Zero W:

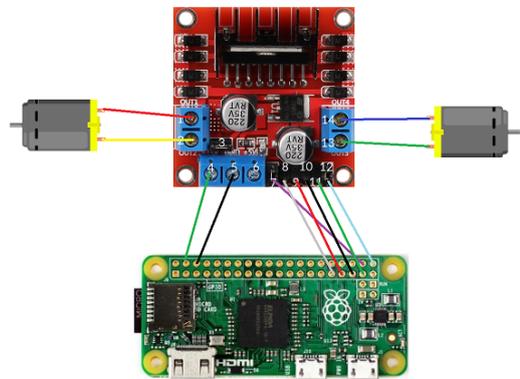


Figure 2: Hardwareaufbau [5]

Die Schrittmotoren sind jeweils rechts und links an der Schrittmotorsteuerung angeschlossen. Der Raspberry Pi Zero W ist über einen 5V sowie GND Pin angeschlossen. Dadurch wird die Schrittmotorsteuerung mit Strom versorgt. Damit die Schrittmotoren reagieren, ist der ENA sowie ENB Anschluss als auch die Inputs 1 bis 4 an die Pins des Raspberry Pi Zero W angeschlossen. Diese werden im Programmcode für die Steuerung definiert. Der ENA Anschluss der Schrittmotorsteuerung ist an GPIO20 des Raspberry Pi angeschlossen. Des Weiteren ist GPIO21 an den ENB Anschluss angebinden. Die Inputs 1 bis 4 sind in folgender Reihenfolge an den Raspberry Pi Zero W angeschlossen: GPIO6, GPIO13, GPIO19 und GPIO26.[5]

Die folgenden Bilder zeigen das fertig gebaute ferngesteuerte Auto mit den Hardwarekomponenten:



Figure 3: Hardwarearchitektur [5]



Figure 4: Hardwarearchitektur [5]

2.2.2 Softwarearchitektur

Als Minicomputer wird der Raspberry Pi Zero W verwendet. Dieser stellt die Schnittstelle da, um die Schrittmotoren zu steuern. Auf dem Raspberry Pi Zero W ist der MJPG-streamer installiert: Mit diesem Werkzeug können die Videodateien der Webcam gestreamt werden. Die Installation des MJPG-streamer orientiert sich an bereits erstellten Anleitungen.[7] In der Datei "input-uvv.so" wird das Format von MJPEG zu YUYV gewechselt. Im MJPEG-Format wird der Stream nicht angezeigt, wodurch ein Wechsel des Formates notwendig ist. Des Weiteren wird der WebIOPi Server installiert. Mithilfe des Webservers kann die Schrittmotorsteuerung durch ein Webinterface gesteuert werden. Dabei wird ebenfalls bereits erstellten Anleitungen gefolgt. Die Datei "config" von WebIOPi verlinkt auf die html-Datei, welche die Buttons für die Steuerung zeigt, sowie auf die Python-Datei, welche die Funktionalität der Buttons definiert.[7]

Wie bereits genannt wird für das Webinterface eine html-Datei benötigt, welche die Buttons für die Steuerung anzeigt. Die Datei rabicarweb.html wird in dem Verzeichnis htdocs abgelegt. Dabei entspricht die Datei rabicarweb.html einem Projekt, wo bereits ein ferngesteuertes Auto mit einem Raspberry Pi und einer Kamera gebaut wurde.[4] Die Datei "config" von WebIOPi verlinkt auf rabicarweb.html. [7] Die Schrittmotorsteuerung kontrolliert die Motoren und definiert die Leistung sowie welche Motoren angesteuert werden. Dieses wird in der Datei L298NHBridge.py implementiert, welche sich im home-Verzeichnis des Raspberry Pi befindet. Der Programmcode orientiert sich mit kleinen Änderungen an

einem bereits erstellten Projekt mit einem ferngesteuerten Auto. [4] Die letzte Datei, welche benötigt wird, definiert die Buttons der html-Datei und steuert die Schrittmotorsteuerung. Diese ist ebenfalls im home-Verzeichnis des Raspberry Pi abgelegt. Ein bereits erstelltes Projekt verwendet diese Datei: RobotControlWeb.py. [4] Die Datei ist für die Verwendung von zwei Motoren angepasst. Mit dieser Datei wird der WebIOPi Server gestartet und das Webinterface kann über die IP-Adresse des Raspberry Pi aufgerufen werden.

Für eine einfache Bedienbarkeit werden die benötigten Programme als Autostart über rc.local durchgeführt.

2.2.3 Test und Verwendung

Für die Inbetriebnahme müssen in die Batteriehalterungen AA Batterien eingelegt werden. Durch die Betätigung des Schalters startet der Raspberry Pi Zero W. Die rote Lampe der Kamera geht nach einiger Zeit an. Dann ist der MJPG-streamer gestartet. Die Datei RobotControlWeb.py wird automatisch gestartet und ermöglicht den Zugriff auf das Webinterface. Der Raspberry Pi Zero W muss mit dem Internet verbunden sein und sich im selben Netzwerk befinden wie die Steuerungskomponente. Dabei kann ein Laptop oder internetfähiges Smartphone verwendet werden. Wenn der Raspberry Pi noch keine Internetverbindung hat, muss diese zunächst hergestellt werden. Die IP-Adresse wechselt, wenn sich der Raspberry Pi mit einem neuen Netzwerk verbindet. Damit der Stream angezeigt werden kann, muss diese in rabicarweb.html hinterlegt werden. Diese Datei befindet sich unter folgendem Pfad: /usr/share/webiopi/htdocs. Sobald eine Inter-

netzverbindung hergestellt ist und sich beide Geräte im gleichen Netzwerk befinden, kann das Webinterface über folgende Adresse aufgerufen werden: `http://IP-Adresse des RaspberryPi:8000/rapicarweb.html`. Die Adresse ist in folgende Bestandteile aufgebaut: Der erste Teil stellt die IP-Adresse des Raspberry Pi da. Darauf folgt der Port, welcher in der Datei `RobotControlWeb.py` definiert ist sowie die `html`-Datei, welche die Buttons und den Stream anzeigt. Im Browser wird folgende Seite angezeigt:

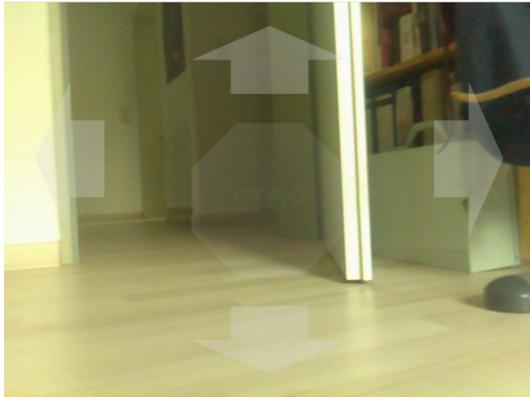


Figure 5: Webinterface [5]

Die Pfeiltasten sind für die Steuerung des Autos zuständig. Im Hintergrund wird der Live-Stream angezeigt. Das Auto kann sowohl in Außen- als auch Innenbereichen verwendet werden. Dabei muss sich das Auto und das Endgerät im gleichen Netzwerk befinden. Sobald ein Gerät nicht mehr verbunden ist, kann das Auto nicht mehr gesteuert werden. Der Nutzer muss sich nicht im gleichen Raum befinden wie das Auto. Durch den Live-Stream kann der Nutzer sehen, wo sich das Auto befindet.

3. Zusammenfassung und Fazit

Das Auto lässt sich über ein Webinterface mit einem Live-Stream im Hintergrund steuern. Somit findet das Auto wieder eine Verwendung und wurde "upgecycelt". Mithilfe eines Raspberry Pi sind die Anwendungsmöglichkeiten erweitert und verbessert worden. Somit war der Prozess des "Upcycling" erfolgreich. Die Handhabung des ferngesteuerten Autos lässt sich noch erweitern. Der Raspberry Pi sowie das Endgerät müssen sich in einem Netzwerk befinden, damit das Webinterface gestartet werden kann. Dadurch ist die Benutzung eingeschränkt und das Auto kann nur in bestimmten Bereichen verwendet werden. Als Ausblick besteht die Möglichkeit den Raspberry Pi als Hotspot einzurichten. Dadurch kann sich der Nutzer mit seinem Endgerät direkt verbinden und das Auto in allen Bereichen steuern. Der Raspberry Pi muss nicht mit einem

Netzwerk verbunden sein.

Des Weiteren können, wie bereits bei den Möglichkeiten zum Upcycling genannt, noch Sensoren eingebaut werden. Ein Bewegungssensor könnte beispielsweise eine Meldung ausgeben, wenn sich eine Person in der Nähe des Autos befindet. Um Entfernungen zu messen, wird ein Ultraschallsensor benötigt. Es bestehen noch weitere Möglichkeiten ein ferngesteuertes Auto aufzuwerten. Im Internet sind einige Projekte zu finden, welche einen bei der Durchführung helfen sowie unterstützen und somit als Grundlage verwendet werden können.

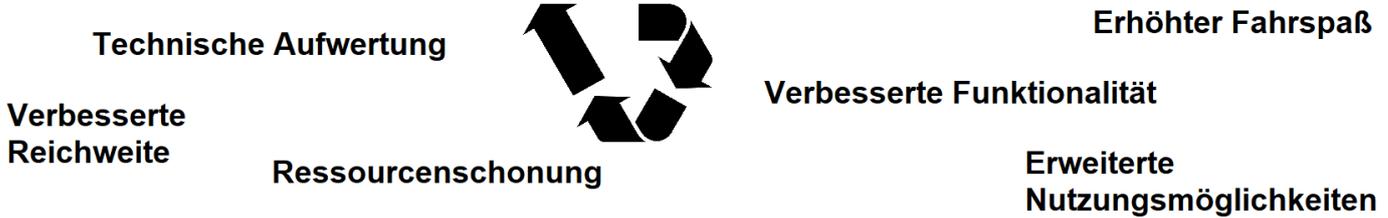
References

- [1] Code Rewind. Raspberry Pi 2 WiFi RC Car. <https://www.instructables.com/id/Raspberry-Pi-2-WiFi-RC-Car/>, 2019. Zugriff am 18.06.2019.
- [2] Dickie Toys. RC Red Thunder, RTR. <https://www.dickietoys.de/de/marken-produkte/rc/landoffroad/rc-red-thunder-rtr-201119121/>, 2019. Zugriff am 18.06.2019.
- [3] ExpertenTesten. Geschichtliche Entwicklung des ferngesteuerten Autos. <https://www.expertentesten.de/ferngesteuertes-auto-geschichtliche-entwicklung/>, 2019. Zugriff am 20.05.2019.
- [4] Ingmar Stapel. Robot Car – programs. <https://custom-build-robots.com/ebook-download?lang=en>, 2019. Zugriff am 18.06.2019.
- [5] JoyIT. Motordriver2. <https://cdn-reichelt.de/documents/datenblatt/X200/SBC-MOTODRIVER2-ANLEITUNG.pdf>, 2019. Zugriff am 20.05.2019.
- [6] J. Kuczera. Heinrich hertz : Entdecker der radiowellen, 1987.
- [7] osoyoooproduct. Raspberry Pi Robot Car Lesson 3: Webcam. <https://www.instructables.com/id/Raspberry-Pi-Robot-Car-Lesson-3-Web-Control-Camera/>, 2019. Zugriff am 18.06.2019.
- [8] K. Sung. A review on upcycling: Current body of literature, knowledge gaps and a way forward. 04 2015.

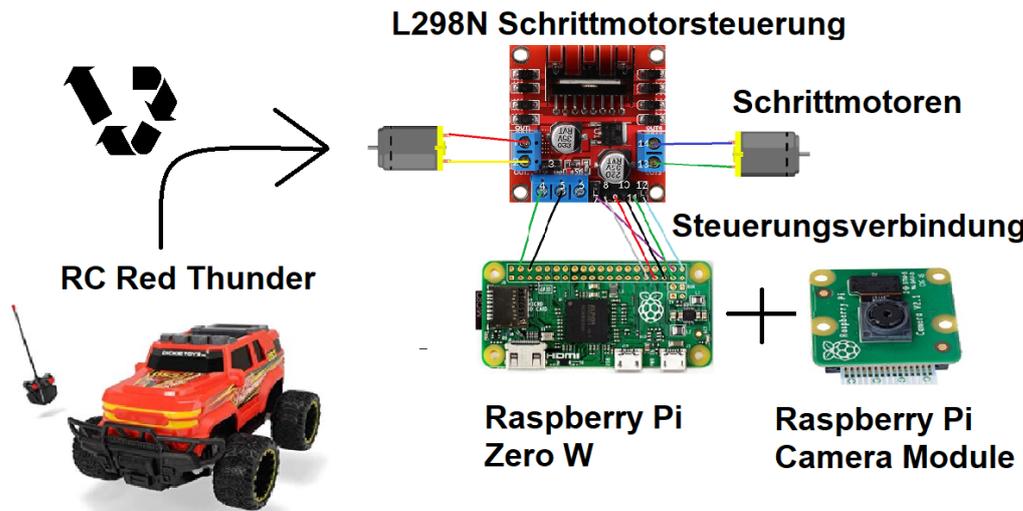
Upcycling eines ferngesteuerten Autos

Paula Bürmann

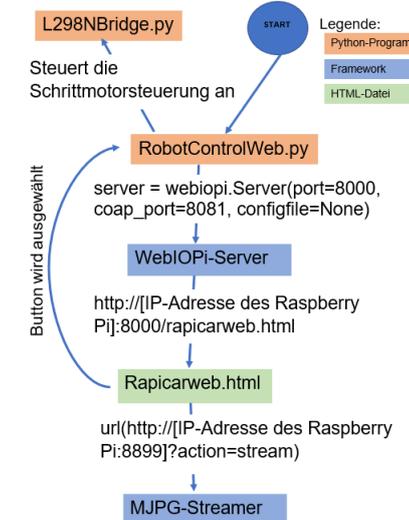
BEGRIFF UPCYCLING



HARDWARE



SOFTWARE



HANDBUCH

• Vorraussetzung

- Endgerät: Handy oder Laptop
- Internetverbindung
- Endgerät und Raspberry Pi im selben Netzwerk
- IP-Adresse des Raspberry Pi

• Webinterface

- [http://\[IP-Adresse des Raspberry Pi\]:8000/rapicarweb.html](http://[IP-Adresse des Raspberry Pi]:8000/rapicarweb.html)



Steuerungskomponenten:

1. Rechts
2. Links
3. Vorwärts
4. Rückwärts
5. Stop

KOSTEN

1. Ferngesteuertes Auto mit Schrittmotoren	12 Euro
2. Raspberry Pi Zero W	15 Euro
3. Schrittmotorsteuerung	6 Euro
4. Batteriehalter	1 Euro
5. Raspberry Pi Kamera Modul	24 Euro
Gesamtkosten	58 Euro

QUELLEN

References

- [1] Reichelt Elektronik, *Motor Driver 2*, "https://cdn-reichelt.de/documents/datenblatt/A300/SBC-MOTORDRIVER2.DATENBLATTV.2.pdf, 2019".
- [2] osoyoproduct, *Raspberry Pi Robot Car Lesson 3: Webcam*, "https://www.instructables.com/id/Raspberry-Pi-Robot-Car-Lesson-3-Web-Control-Camera".
- [3] Dickie Toysa, *RC Red Thunder, RTR*, "https://www.dickietoys.de/marken-produkte/rc/landoffroad/rc-red-thunder-rtr-201119121/".