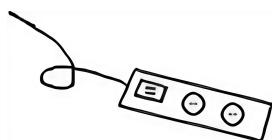


# Smarte Steckdosenleiste

Jonathan Westphal (16)



**Fachgebiet:** Technik

**Bearbeitungsort:** Karlsruhe

**Wettbewerbssparte::** Jugend Forscht

**Bundesland:** Baden-Württemberg

**Projektbetreuende:** Keine

**Wettbewerbsjahr:** 2025

# Contents

<b>1 Kurzfassung</b>	<b>2</b>
<b>2 Motivation und Fragestellung</b>	<b>2</b>
2.1 Wer bin ich? . . . . .	2
2.2 Wie kam ich auf die Idee? . . . . .	2
2.3 Was habe ich erarbeitet? . . . . .	3
<b>3 Hintergründe und Erkenntnisse</b>	<b>4</b>
3.1 Einsparpotenzial Standby Verbrauch . . . . .	4
3.2 Überspannungsschutz . . . . .	4
3.3 "Softanlauf" zur Vermeidung von Einschaltspitzen . . . . .	5
3.4 Leistungsmessung . . . . .	7
3.5 "Smarte" Funktionen . . . . .	9
3.6 Unterscheidung & Alleinstellungsmerkmale . . . . .	10
<b>4 Aufbau und Vorgehensweise</b>	<b>11</b>
4.1 Entwicklung des Schaltplan und Auswahl der elektronischen Bauteile . . . . .	11
4.2 Design und Layout der Platine . . . . .	12
4.3 Bestücken der Bauteile & Inbetriebnahme . . . . .	13
4.4 Test und Programmierung . . . . .	14
<b>5 Ergebnis</b>	<b>17</b>
5.1 Umweltschutz ( $\text{CO}_2$ ) . . . . .	17
5.2 Kosten Nutzen Rechnung . . . . .	17
<b>6 Fazit</b>	<b>18</b>
<b>7 Datenblätter</b>	<b>19</b>
7.1 Erste Version der Steckdosenleiste . . . . .	19
7.2 Zweite Version der Steckdosenleiste . . . . .	19

# 1 Kurzfassung

In der selbst entwickelten und hergestellten "Smarten Steckdosenleiste" habe ich die Funktionen: Schutz von elektrischen Geräten, Softanlauf (Vermeidung von hohen Einschaltströme), Leistungsmessung und "smarte" Funktionen, wie ein Webinterface, integriert. Aus diesen Gründen dient sie einer kostengünstigen Realisierung zur Stromeinsparung und trägt zum Umweltschutz bei.

Meine wissenschaftlichen Erkenntnisse waren, dass ich je nach elektrischem Verbraucher bei der Bestimmung eines Standby Stromverbrauches zwischen Wirkleistung, Blindleistung und Scheinleistung differenzieren muss. Hierzu musste ich Spannung und Strom messen und deren Phasenverschiebung bestimmen.

## 2 Motivation und Fragestellung

### 2.1 Wer bin ich?

Ich bin ein 16-jähriger technikbegeisterter Jugendlicher und besuche derzeit das Bismarck-Gymnasium in Karlsruhe.

In den vergangenen Jahren habe ich mich intensiv mit Elektronik beschäftigt und konnte dabei bereits wertvolle Erfahrungen sammeln. So habe ich am CanSat-Wettbewerb<sup>1</sup> teilgenommen und dabei den zweiten Platz erreicht. Darüber hinaus habe ich zahlreiche eigene Elektronikprojekte zu Hause entwickelt und erfolgreich umgesetzt, darunter ein Fahrradbremslicht, sowie eine Wetterstation mit Serveranbindung.

### 2.2 Wie kam ich auf die Idee?

Die Idee für dieses Projekt entstand bei dem Blick auf meinen Schreibtisch. Hier sind viele verschiedene elektronische Geräte angeschlossen (PC, Monitor, Laptop, Labornetzteil, Lötkolben, Receiver, Lautsprecher, diverse Ladegeräte ...). Dabei hatte ich folgende Gedanken:

- Status LEDs brennen dauerhaft
- Standby Stromverbrauch auch bei ausgeschaltetem Gerät

---

<sup>1</sup>Deutscher Wettbewerb zum Bau eines Minisatellitenbau in Dosenformat ([cansat.de](http://cansat.de))

- Steckdosenleiste mit unzureichendem Überspannungsschutz
- Welchen Energie- / Stromverbrauch besitzen meine Geräte

Anfangs hatte ich das Problem dadurch gelöst, dass ich eine schaltbare Steckdosenleiste verwendete. Doch mit der Zeit vielen mir weitere Probleme auf:

- Unnötiges Warten auf das Abschalten der Steckdosenleiste, insbesondere wenn am PC noch ein Update läuft
- Spitzenströme beim Einschalten führten zur defektem Schalter in der Steckdosenleiste und zur Auslösung der Haushaltssicherung. Diese musste ich dann immer zwei Stockwerke tiefer im Keller wieder einschalten. Hinzu kam, dass ich diverse elektronische Geräte wieder einrichten musste (Bsp. elektronische Uhr).

Deshalb beschloss ich eines Tages, dass ich genug in den Keller gegangen bin. Es musste eine ordentliche Lösung her.

## 2.3 Was habe ich erarbeitet?

So entstand die Idee der smarten Steckdosenleiste. Doch welche Funktionen sollte diese haben? Zuerst habe ich festgelegt, dass sie erkennen muss, wenn ich mich vom Schreibtisch entferne. Am einfachsten und universellsten ließ sich das durch eine Leistungsmessung des Stromverbrauchs realisieren. Außerdem sollte die Steckdosenleiste die Möglichkeit haben, beim Einschalten kurzzeitig einen Widerstand zuzuschalten. Dadurch wird vermieden, dass die Sicherung durch den plötzlichen Einschaltstrom auslöst. Als weiteres Feature hatte ich die Idee, die Steckdosenleiste in ein Smart-Home-Netzwerk integrieren zu können, um sie von überall aus steuern zu können und damit es einfacher ist beim Entwickeln einen neuen Code hochzuladen. Als letztes entschied ich mich dafür, Sicherheitsfeatures einzubauen, die meiner Meinung nach jede Steckdosenleiste haben sollte: ein Überspannungsschutz für Blitzschläge und eine Backup Schmelzsicherung.

In der selbst entwickelten und hergestellten "Smarten Steckdosenleiste" habe ich folgende Funktionen entwickelt:

**Eliminierung von Standby Verbrauch** bei angeschlossenen Geräten

**Schutz** von elektrischen Geräten vor Überspannung

”**Softanlauf**“ - hier geht es darum hohe Einschaltströme zu vermeiden.  
Diese sind bauartbedingt insbesondere bei Schaltnetzteilen zu finden.

**Leistungsmessung** mit der Möglichkeit, zwischen Wirkleistung, Blindleistung und Scheinleistung zu differenzieren.

”**smart**“ - **Möglichkeit** der Integration in eine Hausautomation und Interaktion mit dem Anwender

### 3 Hintergründe und Erkenntnisse

#### 3.1 Einsparpotenzial Standby Verbrauch

Laut einer Statistik des deutschen Umweltbundesamtes<sup>2</sup> verbraucht der durchschnittliche deutsche Haushalt über 320kw/h im Jahr nur für den Standbyverbrauch seiner Geräte. Bei den derzeitigen Energiekosten (0,33 €/kWh) sind das in etwa 100 € im Jahr

$$40 \text{ Watt} \cdot 41,3 \text{ Mill. Haushalte} = 1,6 \text{ Gigawatt}$$

Das entspricht in etwa der Leistung von drei Kohlekraftwerken oder 200 Windrädern. Hierbei sind Standby Verbraucher im Dienstleistungsgewerbe und in der Industrie nicht eingerechnet.

#### 3.2 Überspannungsschutz

Zusätzlich zur Standby-Strom-Eliminierung hat meine Steckdosenleiste einen eingebauten Überspannungsschutz. Gründe für die Überspannung können sein:

**Schalthandlungen** im Stromversorgungsnetz mit ca. 6kV

**Blitzeinschlag** je nach Entfernung und Leitungsart rechnet man mit ca. ~100kV

---

<sup>2</sup>Statistik des Umweltbundesamtes (S.27)

Die Überspannungsschäden belaufen sich in Deutschland auf rund 170 Millionen€.<sup>3</sup>

Alle in meiner Steckdosenleiste eingesteckten Geräte sind mehrstufig absichert:

1. *Überspannung mit einem Varistor* Dies ist ein spannungsabhängiger Widerstand, der mit zunehmender Spannung einen kleineren Widerstand annimmt ⇒ im Extremfall die Überspannung kurzschließt und gegen den N-Leiter ableitet.
2. *Galvanische Trennung* durch die Entkopplung mit Hilfe von Relais. Hierbei befinden sich je ein Relais auf dem N- und L-Leiter der 230V~ Netzspannung. Die Relais haben je die Kennwerte von  $1\text{M}\Omega$  und 1kV Trennspannung
3. Zum allgemeinen Schutz ist noch eine Schmelzsicherung (Typ FastBlow⇒ "schnelle" Auslösung) mit 12 A eingebaut.

Unterm Strich sind das bei einem direkten Blitzeinschlag sicherlich keine ausreichende Absicherungen. Jedoch trägt jede einzelne dazu bei, mögliche Schäden durch Überspannung zu reduzieren.

### 3.3 "Softanlauf" zur Vermeidung von Einschaltspitzen

In der Regel sind die Mehrzahl der elektrischen Versorger mit einem Netzteil ausgestattet. Die trifft insbesondere auf Geräte zu, die eine geringe Spannung ( $<24\text{ V}$ ) benötigen. Diese Netzteile zeichnen sich dadurch aus, dass beim Einschalten die Eingangskondensatoren geladen werden müssen. Die Eingangskondensatoren kann man sich als kleine Batterie verstehen, die beim Einschalten mit einem nahezu unbegrenzten Strom ⇒ Stromstoß geladen werden (Abbildung: 1).

---

<sup>3</sup>[Gesamtverband der Deutschen Versicherungswirtschaft \(GDV\)](#)

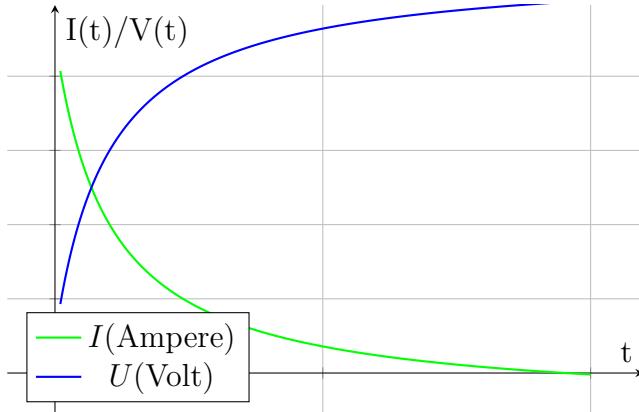


Figure 1: Ladekurve eines Kondensators

Nach meinen Erfahrungen sind dies je nach Phasenanschnitt bei 50Hz mehr als 100 Ampere. Dies führte bei meinen zahlreichen Versuchsaufbauten dazu, dass ich eine Vielzahl an Relais und Schalter durch das Verschweißen der Schaltkontakte zerstört habe.

Beim Nachlesen bin ich dann auf sogennate "High Inrush" Relais gestoßen, die durch Ihre spezielle Beschichtung aus Silber-Zinnoxid ( $\text{AgSnO}_2$ ) Ströme von bis zu 100 Ampere ohne Zusammenkleben aushalten können.

Zusätzlich verwende ich spezielle NTC-Thermistoren, um den Einschaltstrom auf maximal 16 Ampere<sup>4</sup> zu begrenzen. Diese temperaturabhängigen Widerstände weisen bei niedrigen Temperaturen einen hohen Widerstand auf, der mit steigender Temperatur abnimmt. Um den Energieverbrauch weiter zu reduzieren, überbrücke ich die Thermistoren später mit einem Relais. Ich habe mich bewusst für den Einsatz von Thermistoren entschieden, da sie selbstregulierend sind. Das bedeutet, dass sie selbst im Falle eines Relaisausfalls aufgrund ihres temperaturabhängigen Widerstands nicht überhitzen oder durchbrennen würden.

mit  $I$  (Strom),  $U$  (Spannung),  $R$  (Widerstand),  $P$  (Leistung)

$$I = \frac{U}{R} \quad 16A \approx \frac{230V}{15\Omega} \quad P = I^2 R \quad 4kW \approx 16A^2 15\Omega$$

Die enorme Leistung von 4kW und der maximale Strom von 16 Ampere stehen jedoch bei den beiden NTC-Thermistoren nur für den Bruchteil

---

<sup>4</sup>Typische Nennstärke einer Haushaltssicherung

einer Sekunde an, so dass diese nicht thermisch zerstört werden (Datenblatt im Quellenverzeichnis Kapitel 6).

### 3.4 Leistungsmessung

Um den Standby Verbrauch der Geräte zu eliminieren musste ich eine Ausschaltschwelle bestimmen, ab welcher meine Steckdosenleiste ausgeschaltet wird. Hierbei stand ich vor zwei Problemen:

1. *Undefinierte Anzahl* an angeschlossenen elektronischen Geräten
2. *Unterschiedlichste Arten an Verbraucher* - diese können kapazitive, induktive oder ohmsche Lasten sein. Hinzu kommt, dass es Geräte gibt, die mit einer Phasenanschnittssteuerung arbeiten.

In einer ersten Version meiner Steckdosenleiste hatte ich nur die Stromstärke gemessen (Abbildung: 2) und angenommen, dass sich diese linear zum Spannungsverlauf verhält. Jedoch musste ich im Vergleich zu verschiedenen Leistungsmessgeräten (umgangssprachlich Strommessern) feststellen, dass dies nicht der Fall war.

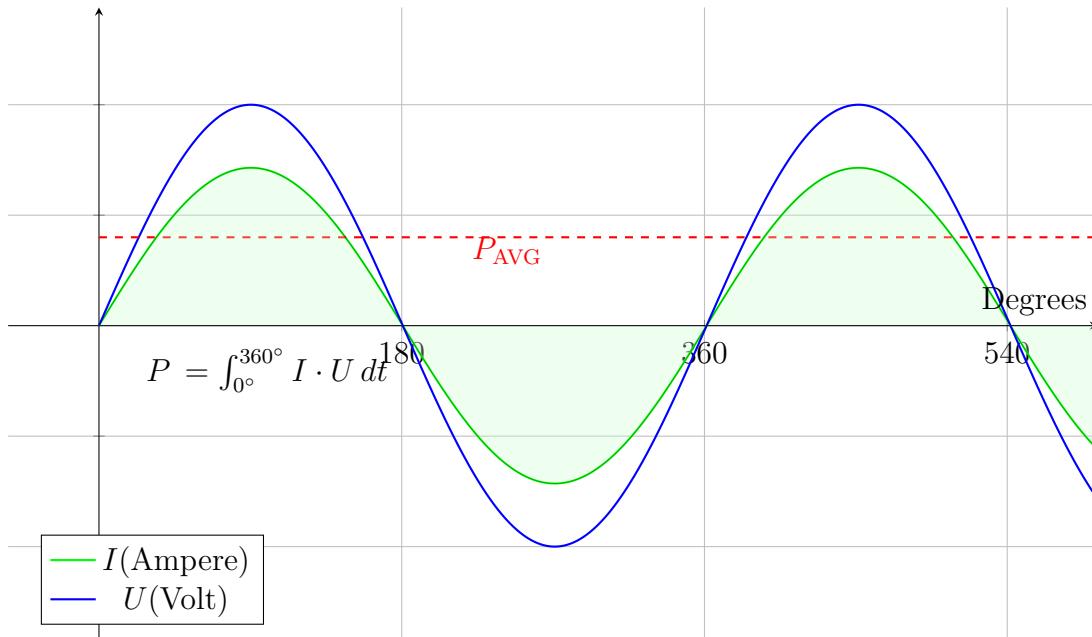


Figure 2: Leistungsmessung nur mithilfe der Stromstärke

Nach vielen ausführlichen Test mit verschiedenen Geräten und Recherche im Internet kam ich zur Erkenntnis: Es gibt zum einen Geräte die sich wie ein Widerstand verhalten  $\Rightarrow$  Spannung und Strom verhalten sich linear. Andererseits gibt es Geräte mit induktiver oder kapazitiver Last  $\Rightarrow$  Spannung und Strom sind phasenversetzt. Dies führt dazu, dass ich zwischen Wirkleistung, Scheinleistung und Blindleistung unterscheiden muss. Entscheidend für die Festlegung der Ausschaltschwelle für den Standby Verbrauch ist das Verhältnis zwischen Wirkleistung und Scheinleistung (PowerFactor):

$$\text{Power Factor} = \frac{\text{Wirkleistung}}{\text{Scheinleistung}} = \cos(\varphi)$$

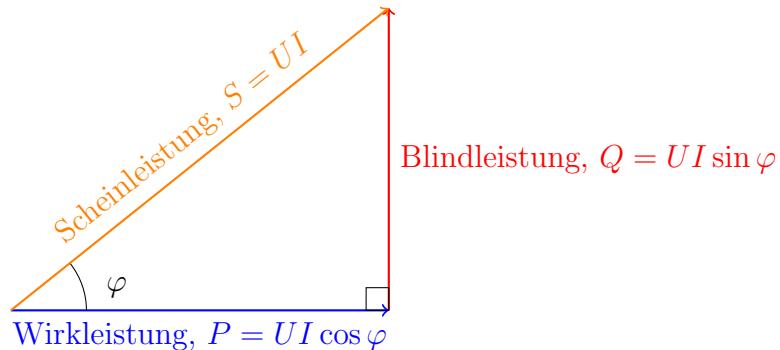


Figure 3: Leistungsdreieck: Zusammenhang zwischen Scheinleistung, Wirkleistung und Blindleistung

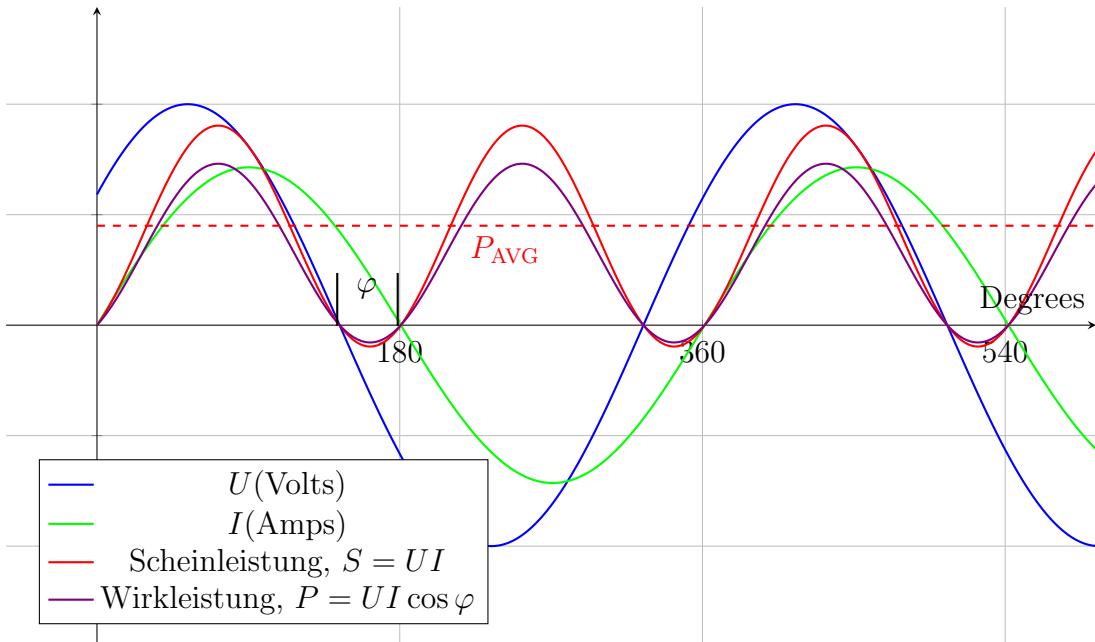


Figure 4: Spannung, Strom und Leistung bei einer Phasenverschiebung

Diese Erkenntnisse habe ich meiner zweiten Version der Steckdosenleiste mit einem modernerem Leistungsmesschip entnommen, der neben dem Strom auch die Spannung und die damit kohärierende Phasenverschiebung misst. Rechnerisch kann ich somit den PowerFactor bestimmen und einen eindeutigen Standby Verbrauch berechnen. (Abbildung: 4)

### 3.5 "Smarte" Funktionen

Auch Smart Home Funktionen habe ich in meiner Anwendung realisiert. Im Gegensatz zu einer handelsüblichen Steckdosenleiste kann meine auch den Verbrauch der Geräte in einem Webinterface detailliert anzeigen und analysieren. Diese Informationen ermöglichen es dem Nutzer seinen Stromverbrauch individuell nachvollziehen zu können. Ebenso gibt es die Möglichkeit für andere Smart Home Geräte mit der Steckdosenleiste zu interagieren. Ein Beispiel hierfür wäre die Solaranlage, die erkennt, dass es keinen Stromüberschuss mehr gibt und in Folge der Steckdosenleiste mitteilt, dass sie sich ausschalten soll.

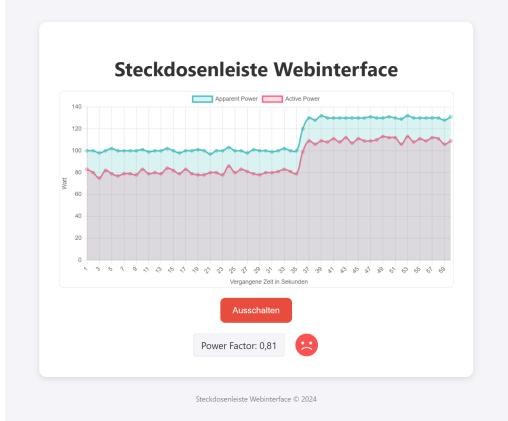


Figure 5: Benutzeroberfläche

```
{
  "status": "angeschaltet",
  "seit": "5min",
  "apparent_power": 131,
  "active_power": 113,
  "reactive_power": 18,
  "power_factor": 0.86
}
```

Figure 6: JSON Ausgabe

### 3.6 Unterscheidung & Alleinstellungsmerkmale

Welche Produkte sind am Markt erhältlichen und wie unterscheidet sich meine Steckdosenleiste?

**MasterSlave Schaltungen** diese schalten nach einer einstellbaren bzw. erlernbaren Schwelle (Master) Verbrauche an einem zweiten Schaltkreis (Slave) ab. Jedoch steht der Master weiterhin unter Strom  $\Rightarrow$  Standby Stromverbrauch und Stromverbrauch der MasterSlave Schaltung.

**Smarte Steckdosen bzw. Managed Server Steckdosen** werden nach vorgegebener Zeit oder Stromverbrauch geschaltet. Da diese jedoch im Netz immer erreichbar sind, verbrauchen sie wiederum Strom (was ich vermeiden wollte). Hinzu kommt eine starre Schwelle für den Standby Verbrauch.

**Schutz Steckdosenleiste** hierbei handelt es sich in der Regel um ein Premiumsegment, dass die angeschlossenen Geräte gegen Überspannung schützen soll. Diese haben jedoch bis auf einen Kippschalter keine weiteren Funktionen.

Die von mir entwickelte smarte Steckdosenleiste bietet als Alleinstellungsmerkmal die komplette Trennung der elektrischen Verbraucher und die vollständige Vermeidung von Standby Verbrauch. Nicht zu vergessen ist der einfache Überspannungsschutz.

## 4 Aufbau und Vorgehensweise

Die Umsetzung erfolgte in mehreren Anläufen mit folgenden Schritten:

- Entwicklung des Schaltplan(Siehe Anhang) und Auswahl der elektronischen Bauteile
- Design und Layout der Platine (Siehe: 7)

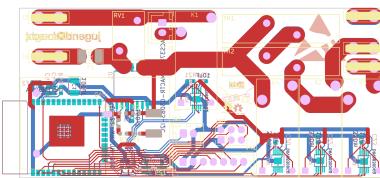


Figure 7: Leiterplattendesign aus KiCad

- Bestücken der Bauteile
- Test und Programmierung

### 4.1 Entwicklung des Schaltplan und Auswahl der elektronischen Bauteile

Den gesamten elektrischen Aufbau, sowie die Schaltpläne habe ich alle mit meinen Vorkenntnissen in KiCad<sup>5</sup> erstellt.

Bei der Auswahl der elektronischen Komponenten will ich folgende Aspekte erläutern:

**SMD** (Surface Mounted Device) Umsetzung auf einem möglichst kleinen Bauraum. Ergänzend können die Komponenten als SMD Ausführung kostengünstig vom Auftragsfertiger JLCPCB<sup>6</sup> zum Teil bestückt werden.

**Bistabile Relais** besitzen die Eigenschaft: mit nur einem Stromimpuls in den jeweiligen anderen Schaltzustand zu wechseln und benötigen daher nicht dauerhaft Strom zum Aufrechterhalten ihrer Leitfähigkeit.

---

<sup>5</sup>Open-Source-Software zur Erstellung von Leiterplatten und elektrischen Schaltplänen ([kicad.org](http://kicad.org))

<sup>6</sup>Kostengünstiger Leiterplattenhersteller in China ([jlcpcb.com](http://jlcpcb.com))

**Prozessor** In meiner ersten Version der Steckdosenleiste habe ich den ESP12-E verwendet, da er für die damaligen Zwecke ausreichend Rechenleistung besaß und kostengünstig, sowie klein war. Für die neuere Version der Steckdosenleiste habe ich mich jedoch dazu entschlossen, den ESP32-WROOM-32D zu verwenden, da er zum einen mehr Pins unterstützt, als auch leistungsstärker ist und somit komplexere mathematische Rechnungen schneller ausführen kann.

**Leistungsmesser** Auch beim Leistungsmesschip habe ich ein "Upgrade" verwendet, das nicht nur die Stromstärke, sondern auch die Spannung und die damit eingehende Phasenverschiebung messen kann. (Siehe Kapitel: 3.4).

**Thermistor** Siehe Kapitel: 3.3.

**Varistor** Bei der Auswahl des Varistors achtete ich vor allem auf die maximale "Surge Current", also auf die maximale Leistung, die er aufnehmen kann bevor er elektrisch zusammenbricht. (Siehe Kapitel: 3.2).

Alle Datenblätter der gerade eben genannten elektrischen Bauteile und der Schaltplan sind im Quellenverzeichnis zu finden.

## 4.2 Design und Layout der Platine

Beim Design der Platinen habe ich die Software KiCad<sup>7</sup> verwendet. Diese erlaubt mir eine Übernahme des Schaltplans und eine visuelle Kontrolle der bestückten Platine.

Insbesondere für den 230V~ Schaltkreis bei gegebenen maximal 16 Ampere habe ich für das Layout der Platine folgende Prämissen berücksichtigt:

**Leiterbahnquerschnitt** je kleiner der Querschnitt ist, desto mehr erwärmt sich die Leiterbahn. Für die korrekte Dimensionierung habe ich daher folgende Formel aus dem Designrichtlinienblatt IPC 2221 genutzt:

$$I = K \cdot \Delta T^{0.44} \cdot (W \cdot H)^{0.725}$$

wobei:

---

<sup>7</sup>Open-Source-Software zur Erstellung von Leiterplatten und elektrischen Schaltplänen ([kicad.org](http://kicad.org))

- $I$  = Maximaler Strom in A
- $\Delta T$  = Temperaturanstieg über die Umgebungstemperatur in °C
- $W$  = Breite in mil
- $H$  = Dicke (Höhe) in mil
- $K$  = Konstanter Faktor 0,048 für außenliegende Leiterbahnen

Der maximale, als unbedenklich betrachtete Temperaturanstieg liegt bei 30°C. Dies ergibt bei den bei mir verwendeten Parameter (Längem, Breite, Dicke) folgende maximale Stromstärke:

$$I = 0,048 \cdot \Delta 30^{0.44} \cdot ((8mm * 40) \cdot (35\mu m / 25))^{0.725}$$

$$I \approx 17,9 \text{ A}$$

Diese liegt somit oberhalb meiner 12 Ampere verbauten Sicherung und der maximalen Haushaltssicherung von 16 Ampere.

**Sicherheitsabstand** von 3,25 Millimeter zwischen dem Niederspannungsbereich (5V/3.3V) des Mikrocontrollers und dem 230~ Netz.

### 4.3 Bestücken der Bauteile & Inbetriebnahme

SMD Bauteile (Widerstände und Kondensatoren) ließ ich vom Hersteller der Platine bestücken. Die restlichen Bauteile verlötete ich mit bleifreiem Lot und überprüfte optisch insbesondere den 230V~ Bereich (persönliche Sicherheitsinteresse).

Dann war es soweit, Schritt für Schritt nahm ich die Platine in Betrieb:

1. Mikrocontroller auflöten und ersten Testsketch hochladen
2. Einbau und Verkabelung des Displays und der Schalter für die Bedienung
3. 230V~ Bereich: Stecker rein und ... es passierte nichts. Zum Glück, es sollte auch nichts passieren, bis auf das regelmäßige Blinken der Status LED des Mikrocontrollers
4. Relais und schalten der elektrischen Verbraucher

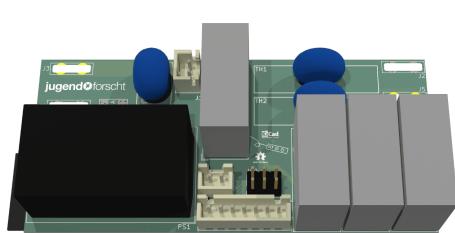


Figure 8: Oberseite der Platine

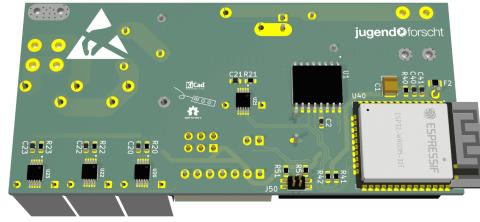


Figure 9: Unterseite der Platine

5. Erste versuche den Leistungsmesschip auszulesen und Strommessungen durchzuführen

Sobald alles "halbwegs" funktionierte baute ich die Schaltung in eine isolierte Steckdosenleiste ein. Dies hatte den Vorteil, dass es keine offenen 230V~ Kontaktstellen mehr gab.



Figure 10: Fertige funktionstüchtige Steckdosenleiste

#### 4.4 Test und Programmierung

Um wenigstens die Programmierung relativ einfach zu halten, habe ich mich entschieden, die gesamte Programmierung in C++ (genauer Arduino C++) zu schreiben. Dabei habe ich für einfache Funktionen, wie z.B. das Anzeigen von Grafiken und Texten auf dem Display Standardbibliotheken verwendet. Den Rest des Codes sowie die Leistungsberechnung habe ich selbst programmiert. Dabei sind mir zwar oft Fehler unterlaufen, aber ich liebte die vielen kleinen Glücksmomente wenn der Code funktioniert hat.

Den eigentliche Programmablauf lässt sich aus dem Flußdiagramm entnehmen. Hierbei unterscheide ich zwischen:

**Hauptroutine** - die ständig Messungen vornimmt und die Werte zur Verfügung stellt, bis die Unterschwelle für Standby Verbrauch erreicht ist.

**Nebenroutine** - die es dem Anwender ermöglicht, über einen Taster weitere Eingaben (Abbruch der Abschaltvorgangs, vorzeitiges Ausschalten, Anzeigen Verbrauchsdiagramm und Codeupdate-Routine) zu machen.

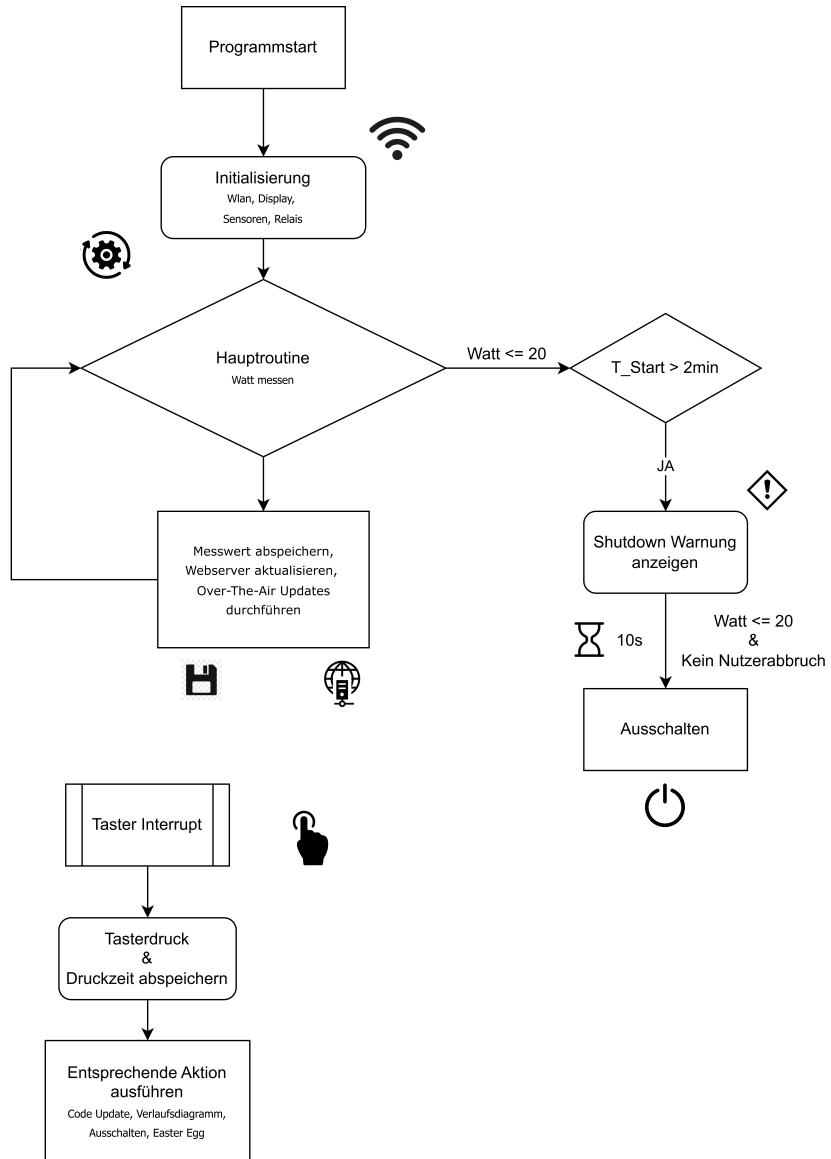


Figure 11: Blockdiagramm des Codes

Viel Zeit gekostet hat mich den korrekten Leistungsverbrauch bei verschiedenen elektrischen Geräten zu messen. Sollte es bei einem Gerät korrekt sein, so war es bei einem weiteren nicht mehr richtig.

## 5 Ergebnis

Aus den zahlreichen Tests und den gewonnenen Erkenntnissen ist die erste Version einer Steckdosenleiste entstanden, die bereits für ein Jahr zuverlässig ihre Dienste ausführt. Mit dieser habe ich dieses Jahr beim "Jugend forscht" Regional Wettbewerb teilgenommen.

Als weiteres Ergebnis musste ich nie wieder in den Keller gehen um die ausgelöste Haushaltssicherung einzuschalten. Was mir viel Frust ersparte.

Aus *wissenschaftlicher Sicht* setze ich mich mit einer genaueren Leistungsmessung auseinander. Hierfür programmierte ich an einem selbstlernendem Algorithmus, der mir Dank Wirk-, Schein- und Blindleistungsmessung ermöglicht, auf verschiedene elektrische Verbraucher zu reagieren.

### 5.1 Umweltschutz ( $\text{CO}_2$ )

Um die komplette Dimension der Vermeidung von Standby Verbrauchen aufzuzeigen möchte ich dies am Beispiel Deutschland machen. Unser elektrische Energiebedarf wird aus 63% erneuerbaren Energie gedeckt<sup>8</sup>. Mit der Annahme, dass pro kWh 360 gr.  $\text{CO}_2$  ausgestoßen werden<sup>9</sup>, lässt sich folgende Gleichung aufstellen:

$$1,6 \text{ Gigawatt} \approx 576 \text{ t } \text{CO}_2$$

(siehe Kapitel 3.1)

Dabei werden 13 Tonnen  $\text{CO}_2$  von einem Hektar Wald gebunden<sup>10</sup>. Somit läge bei einer 100 prozentigen Umsetzung des Standby-Verbrauchs das Einsparpotential in  $\text{CO}_2$  in der Größenordnung des Bayrischen Waldes.

### 5.2 Kosten Nutzen Rechnung

Das Einsparpotenzial pro smarter Steckdosenleiste wollte ich hier verdeutlichen:

$$0,02 \text{ kW} \cdot 20 \text{ Stunden} \cdot 365 \text{ Tage} \cdot 0,33 \text{ €} = 48,18 \text{ €/Jahr}$$

---

<sup>8</sup>-Statistisches Bundesamt von 2024 3. Quartal

<sup>9</sup>-Statistica 2025

<sup>10</sup>-Naturschutzbund Traunstein

Extrakosten durch den elektrischen Energieverbrauch meiner Steckdosenleiste:

$$0,002 \text{ kW} \cdot 4 \text{ Stunden} \cdot 365 \text{ Tage} \cdot 0,33 \text{ €} = 0,96 \text{ €/Jahr}$$

Bereits nach 1,2 Jahren haben sich somit die Anschaffungskosten der Steckdosenleiste amortisiert, und in jedem weiteren Jahr spart man 47,22 €.

Folgende Annahmen habe ich für die Rechnung getroffen:

- Die angeschlossenen Geräte werden jeden Tag 4h lang genutzt
- 0,33 € sind die Kosten je KW/h
- 20 Watt ist der Standby Verbrauch der Geräte
- 2 Watt Eigenverbrauch der Steckdosenleiste im Aktivzustand

## 6 Fazit

Als ich das erste Mal über zwei Stockwerke in den Keller lief um die Sicherung einzuschalten hätte ich nie gedacht, dass mich eine Lösung so sehr beschäftigen würde. Mein Interesse war damals geweckt. Mit jedem Bauteil was ich verlötete und mit jeder Programmroutine die ich schrieb erlangte ich neue Erkenntnisse und neue Fragen.

Nach dem Lesen von einer schier unendlichen Anzahl an Spezifikationen für elektrische Bauteile & Anlagen, dem Erlernen der Grundlagen für Leistungsmessung und der nie endenden Welle an Programmfehlern . . . ich glaube ich habe es geschafft: *endlich gibt es eine bezahlbare, umweltbewusste Lösung elektrische Verbraucher selbstständig auszuschalten und diese gleichzeitig zu schützen.*

# References

## 7 Datenblätter

### 7.1 Erste Version der Steckdosenleiste

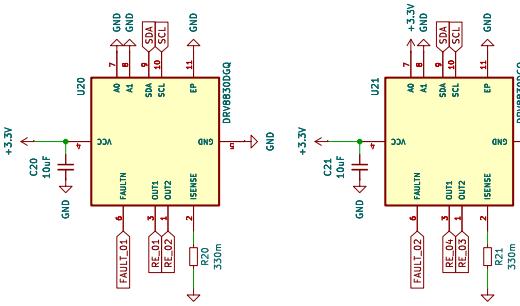
- [1] Microcontroller: ESP12-E (lcsc.com)
- [2] Power Monitoring IC: ACS712 (allegromicro.com)

### 7.2 Zweite Version der Steckdosenleiste

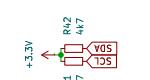
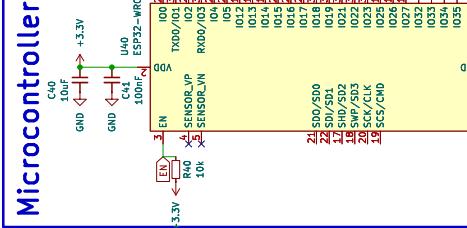
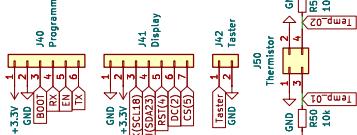
- [3] Microcontroller: ESP32 WROOM 32D (espressif.com)
- [4] Power Monitoring IC: ACS37800 (allegromicro.com)
- [5] Relais Treiber IC: DRV8830 (ti.com)
- [6] High Inrush Relais: ADW1103HTW (industry.panasonic.com)
- [7] NTC-Thermistor: MS32 7R015 (ametherm.com)
- [8] Fast Blow Fuse: 0679H9120-05 (belfuse.com)
- [9] Rückstellbare Sicherung für den Microcontroller und die Relais: MF-PSMF075X-2 (bourns.com)
- [10] Varistor: VDR-10D561K3.5 (lcsc.com)
- [11] Oled Display: Driver ST7735S mit Arduino Standard Bibliothek "TFT eSPI" (docs.arduino.cc)

# 230V

## Relais Treiber



## Interface



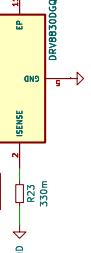
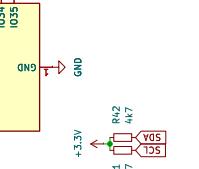
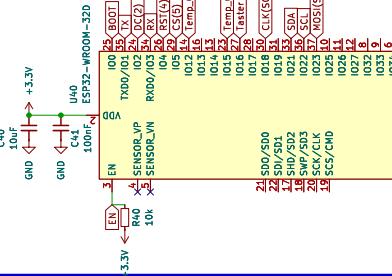
Sheet: /  
File: V0.9.kicad\_sch  
Title: Steckbausenkreisliste 2.0

Size: 33  
KiCad EDA, 9.0.0  
Date: 1/1

Rev: 0.9  
id: 8355888-9e9ea4362ac44cd4d3fe54d26612556131c41b385bbfe624f47ed69

V1.0 Beta  
<https://easyeda.com/editor?id=8355888-9e9ea4362ac44cd4d3fe54d26612556131c41b385bbfe624f47ed69>

## Microcontroller



Sheet: /  
File: V0.9.kicad\_sch  
Title: Steckbausenkreisliste 2.0

Size: 33  
KiCad EDA, 9.0.0  
Date: 1/1

Rev: 0.9  
id: 8355888-9e9ea4362ac44cd4d3fe54d26612556131c41b385bbfe624f47ed69

