

# Videoausgabe mit Mikro- controllern

Von  
Composite-Video  
zu DVI

S. 6

S. 70

## THE TUBE

Ein Röhrenverstärker  
der anderen Art

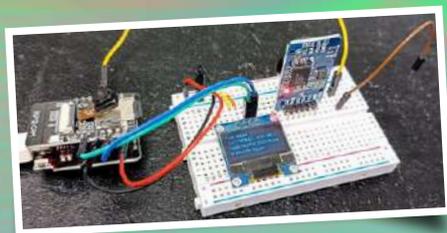
S. 16

## ESP32-KAMERA

So einfach, dass sie nicht  
einmal WLAN braucht

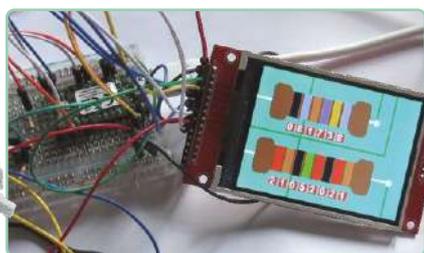
IM FOKUS

# Audio & Video



ATX-Netzteil für Raspberry Pi:  
Verschiedene Spannungen und  
interessante Ströme

S. 20



SDR-Funkuhren:  
Fünf Zeitzeichen, sechs  
Anzeigen

S. 29



Licht für Klangeffekte nutzen:  
LDR-basiertes spannungs-  
gesteuertes Synthesizer-Filter

S. 98





# UNSER SORTIMENT VON TECHNIKERN FÜR TECHNIKER

The best part of your project: [www.reichelt.de](http://www.reichelt.de)

## Nur das Beste für Sie – von über 1.500 Markenherstellern

Unsere Produktmanager sind seit vielen Jahren bei reichelt tätig und kennen die Anforderungen unserer Kunden. Sie stellen ein breites Spektrum an Qualitätsprodukten zusammen, optimal auf den Bedarf in Forschung & Entwicklung, Instandhaltung, IT-Infrastruktur und Kleinserienproduktion sowie auf Maker zugeschnitten.

## STRUCTURE ENERGY: ENERGIE EFFIZIENT STEUERN

*Shelly*

### Shelly TRV Heizkörperthermostat

#### WLAN getauertes thermostatisches Heizkörperventil

Hocheffizientes Heizungsmanagement zur Einsparung von Energiekosten. Durch den großen Temperaturbereich können Sie die Temperatur genau auf Ihre Bedürfnisse einstellen.

- in knapp zwei Minuten gegen das alte Heizungsventil getauscht
- 6500 mAh Akku
- bis zu zwei Jahre Akkukapazität
- Temperaturbereich: 5°C - 30°C



### PRODUKT-TIPP

Bestell-Nr.:  
SHELLY TRV

**79,<sup>90</sup>**

Entdecken Sie unser umfangreiches Shelly-Sortiment für maximalen Wohnkomfort

► [www.reichelt.de/shelly](http://www.reichelt.de/shelly)



SMART. OPTIMAL. PRAKTISCH.

## ENERGIE EFFIZIENT STEUERN

Jetzt entdecken ►  
<https://rch.lt/energy>



**reichelt**  
elektronik **MAGAZIN**

### RATGEBER:

Effizientes Heizen leicht gemacht –  
Mit diesen Tipps lassen sich  
Energie und Kosten sparen

Jetzt entdecken ►  
<https://rch.lt/MG530>



■ Top Preis-Leistungs-Verhältnis

■ über 130.000 ausgesuchte Produkte

■ zuverlässige Lieferung – aus Deutschland in alle Welt

[www.reichelt.de](http://www.reichelt.de)

Bestellhotline: +49 (0)4422 955-333

**reichelt**  
elektronik – The best part of your project

Es gelten die gesetzlichen Widerrufsregelungen. Alle angegebenen Preise in € inklusive der gesetzlichen MwSt., zzgl. Versandkosten für den gesamten Warenkorb. Es gelten ausschließlich unsere AGB (unter [www.reichelt.de/agb](http://www.reichelt.de/agb), im Katalog oder auf Anforderung). Abbildungen ähnlich. Druckfehler, Irrtümer und Preisänderungen vorbehalten. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande, Tel.: +49 (0)4422 955-333

TAGESPREISE! Preisstand: 14.12. 2022

#### Verlag

Elektor Verlag GmbH  
Kackertstraße 10  
52072 Aachen  
Tel. 0241 95509190

Technische Fragen bitten wir per E-Mail  
an [redaktion@elektor.de](mailto:redaktion@elektor.de) zu richten.

#### Hauptsitz des Verlags

Elektor International Media  
Postbus 11, 6114 ZG Susteren  
Niederlande

#### Anzeigen

Margriet Debeij (Leitung)  
Mobil: +31 6 380 780 29  
E-Mail: [margriet.debeij@elektor.com](mailto:margriet.debeij@elektor.com)

Büsra Kas

Tel. 0241 95509178  
E-Mail: [busra.kas@elektor.com](mailto:busra.kas@elektor.com)

Es gilt die Anzeigenpreisliste ab 01.01.2022.

#### Distribution

IPS Pressevertrieb GmbH  
Postfach 12 11, 53334 Meckenheim  
Tel. 02225 88010

#### Druck

Senefelder Misset (Niederlande)  
Mercuriusstraat 35, 7006 RK Doetinchem

Der Herausgeber ist nicht verpflichtet, unverlangt eingesandte Manuskripte oder Geräte zurückzusenden. Auch wird für diese Gegenstände keine Haftung übernommen. Nimmt der Herausgeber einen Beitrag zur Veröffentlichung an, so erwirbt er gleichzeitig das Nachdruckrecht für alle ausländischen Ausgaben inklusive Lizenzen. Die in dieser Zeitschrift veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen einschließlich Platinen sind urheberrechtlich geschützt. Ihre auch teilweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet. Die veröffentlichten Schaltungen können unter Patent- oder Gebrauchsmusterschutz stehen. Herstellen, Feilhalten, Inverkehrbringen und gewerblicher Gebrauch der Beiträge sind nur mit Zustimmung des Verlages und ggf. des Schutzrechtsinhabers zulässig. Nur der private Gebrauch ist frei. Bei den benutzten Warenbezeichnungen kann es sich um geschützte Warenzeichen handeln, die nur mit Zustimmung ihrer Inhaber warenzeichengemäß benutzt werden dürfen. Die geltenden gesetzlichen Bestimmungen hinsichtlich Bau, Erwerb und Betrieb von Sende- und Empfangseinrichtungen und der elektrischen Sicherheit sind unbedingt zu beachten. Eine Haftung des Herausgebers für die Richtigkeit und Brauchbarkeit der veröffentlichten Schaltungen und sonstigen Anordnungen sowie für die Richtigkeit des technischen Inhalts der veröffentlichten Aufsätze und sonstigen Beiträge ist ausgeschlossen.

## von Jens Nickel

Chefredakteur ElektorMag



# Für Augen und Ohren

Die Mischung macht's - seit jeher bei Elektor. Beim Eintauchen in einen Schaltplan kann man genauso etwas lernen wie beim Vertiefen in Quellcode. Auch dem attraktiven Thema Audio & Video nähern wir uns sowohl analog als auch digital, in Theorie und Praxis. Denn schließlich wollen nicht nur die Augen und die Ohren auf ihre Kosten kommen, sondern auch die graue Masse dazwischen.

Unser Cover-Artikel lädt besonders zum Mitmachen ein: Ein günstiges Controllerboard, ein paar externe Bauteile und etwas Open-Source-Software genügen, um Videosignale auszugeben, die man auf einem Monitor betrachten kann. Das klappt nicht nur mit einem Video-Composite-Signal, auch VGA und HDMI sind möglich. Mein Kollege Mathias Claußen hat einen Artikel voller Hintergrundwissen zusammengestellt, der dann allerdings so umfangreich wurde, dass er sogar unser Redaktionssystem an den Rand des Möglichen brachte. Statt in dieser Ausgabe einen 20-seitigen Artikel abzudrucken, haben wir uns entschlossen, das Ganze in zwei Teile zu teilen. In dieser Ausgabe geht es darum, wie man einem kleinen 8-bit-Controller ein Composite-Signal entlocken kann, in Schwarz/Weiß, als Graustufenbild oder in Farbe (Seite 6). In der nächsten Ausgabe sind dann etwas stärkere Rechenkünstler im Einsatz, um VGA-, DVI- und HDMI-Signale zu generieren. Fortgeschrittenen möchte ich den Artikel von Tam Hanna ans Herz legen, der uns das Audio Development Framework von Espressif vorstellt - und zwar anhand praktischer Applikationen wie einem MP3-Player (Seite 88). Ich habe gestaunt, was mit einem ESP32 und ein paar Zeilen Software in punkto Audioverarbeitung alles möglich ist. Zugegeben, die Lernkurve ist steil. Doch für alle diejenigen, die professionelle Projekte mit Sprach- oder Musikausgabe entwickeln möchten, dürfte sich das „Durchbeißen“ in jeder Hinsicht auszahlen.

Ganz ohne Mikrocontroller kommt unser Verstärker für 32-Ω-Kopfhörer aus. Ein Dual-Audio-Opamp OPA2134 wird von zwei BUF634A ergänzt, welche die Stromverstärkung übernehmen. Wer mag, kann diese ICs durch eine diskrete Schaltung ersetzen. Machen Sie mit, lernen Sie dazu!

## — Unser Team —



**Chefredakteur:** Jens Nickel (v.i.S.d.P.) ([redaktion@elektor.de](mailto:redaktion@elektor.de))  
**Redaktion:** Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf (RG, SG), Alina Neacsu, Dr. Thomas Scherer, Brian T. Williams  
**Elektor-Labor:** Mathias Claußen, Ton Giesberts, Clemens Valens  
**Grafik & Layout:** Harmen Heida, Sylvia Sopamena, Patrick Wielders  
**Herausgeber:** Erik Jansen

DEUTSCHE

FACHPRESSE

Elektor ist Mitglied des 1929 gegründeten VDZ (Verband Deutscher Zeitschriftenverleger), der „die gemeinsamen Interessen von 500 deutschen Consumer- und B2B-Verlagen vertritt.“

## Videoausgabe mit Mikrocontrollern

### Teil 1: Composite Video

6



## Rubriken

- 3 Impressum**
- 34 Aller Anfang ...**  
muss nicht schwer sein: Spezielle Dioden
- 38 Aus dem Leben gegriffen**  
Über die Qualität der Dinge
- 52 Von Entwicklern für Entwickler**  
THD-Messung mit Oszilloskop und FFT
- 77 Ethics in Action**  
Biomaterial in der Elektronik: Bereit oder nicht?
- 103 Retronik**  
Hochleistungsverstärker Gigant
- 106 Das besondere Projekt**  
Ein volumetrisches Display Made in Canada
- 110 Projekt 2.0**  
Korrekturen, Updates und Leserbriefe
- 114 Hexadoku**  
Sudoku für Elektroniker

- 47 MakePython ESP32 Development Kit**  
Alles in einer Box
- 80 Opera-Cake-Antennenumschalter für HackRF One**  
Schließen Sie bis zu acht Antennen an Ihr SDR an
- 82 Interview**  
Technik mit Arduino und mehr
- FOKUS**
- 88 Audio mit dem ESP32**  
Das Framework ESP-ADF in der Praxis

## Industry

- FOKUS**
- 56 Allsehende Maschinen**  
Die Technologie hinter modernen industriellen Bildverarbeitungssystemen
- FOKUS**
- 60 Infografik**  
Fakten und Zahlen
- FOKUS**
- 62 Die Entwicklung der Sprach- und Audiosteuerung für elektronische Geräte**
- 66 WEEF 2022 im Rückblick**
- 68 Im Rückblick: FFWD electronica 2022**  
Innovatoren beeindruckten weiterhin!

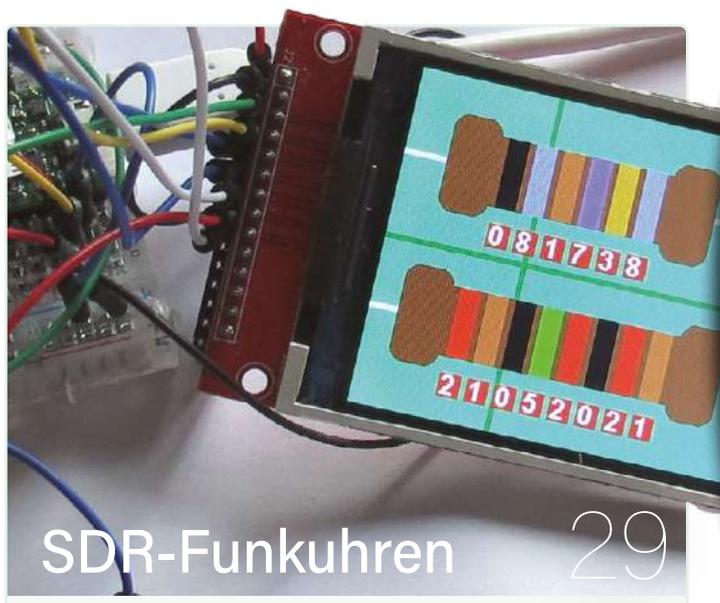
## Hintergrund

- FOKUS**
- 6 Videoausgabe mit Mikrocontrollern**  
Teil 1: Composite Video
- 14 electronica 2022**  
Neuigkeiten von der Weltleitmesse

**The Tube**  
Ein Röhrenverstärker der anderen Art



70



SDR-Funkuhren 29



Licht für Klangeffekte nutzen 98

## Projekte

- FOKUS**  
16 **ESP32-Kamera**  
So einfach, dass sie nicht einmal WLAN braucht
- 20 **ATX-Netzteil für Raspberry Pi**
- FOKUS**  
26 **32-Ω-Kopfhörerverstärker**  
Einfach, aber hohe Qualität mit drei Chips
- 29 **SDR-Funkuhren**
- 40 **Reverse-Engineering eines LED-Displays mit Bluetooth LE**  
Wie man ein BLE-Gerät mit einem Python-Skript steuert
- FOKUS**  
70 **The Tube**  
Ein Röhrenverstärker der anderen Art
- 85 **LiDAR-Präzisionsmetermaß**  
Misst bis zu zwölf Meter
- FOKUS**  
98 **Licht für Klangeffekte nutzen**  
LDR-basiertes spannungsgesteuertes Synthesizer-Filter

## Vorschau

### Elektor März/April 2023

Das nächste Heft ist wie immer randvoll gefüllt mit Spannungsjahrenprojekten, Grundlagen sowie Tipps und Tricks für Elektroniker. Schwerpunkt der nächsten Ausgabe ist Audio- und Videoelektronik.

#### Aus dem Inhalt:

- > Alarm Clock 2.0
- > Videoausgabe mit Mikrocontrollern: VGA, HDMI, DVI
- > Raspberry Pi Pico: PIO in der Praxis
- > Programmierung spannungsgesteuerter IoT-Applikationen
- > Die Android-WiFi-API
- > Mein erstes Software Defined Radio
- > ChipTweaker für Arme
- > Echter Zufallszahlen-Generator mit USB

#### Und vieles mehr!

Elektor **März/April 2023** erscheint am 16. März 2023.

47

**MakePython ESP32 Development Kit**  
Alles in einer Box



# Videoausgabe mit Mikrocontrollern

## Teil 1: Composite Video



Von Mathias Claußen (Elektor-Labor)

Das Thema „Videoausgabe bei Mikrocontrollern“ reicht bis in die Anfänge dieser kleinen Allround-Chips zurück. Heutige Mikrocontroller haben erheblich mehr Rechenleistung als zum Beispiel der fast 42 Jahre alte „Heimcomputer“ Sinclair ZX81, doch von den in GigaByte gemessenen Speichern moderner Grafikkarten sind auch aktuelle Mikrocontroller weit entfernt. Dennoch gelingt es Entwicklern immer wieder, mit einem ATmega, ESP32 oder RP2040 erstaunliche bewegte Bilder auszugeben. In diesem ersten Teil beschäftigen wir uns mit der Ausgabe von Composite Video, im nächsten Heft sind VGA und sogar DVI an der Reihe. In jedem Fall sind ein paar Tricks und ein exaktes Timing nötig. Daher geht es nicht nur um die Theorie, sondern auch um praktische Beispiele als Ausgangsbasis für eigene Experimente.

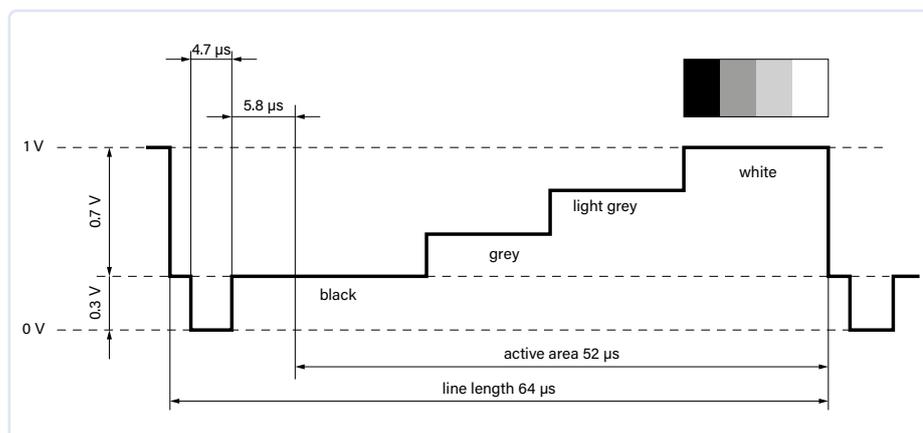


Bild 1. BAS-Signal mit PAL-Timing (Quelle: Wikipedia).

Die Geschichte der Videoformate reicht weit zurück in die Anfänge des Fernsehens. Wie bei Radioübertragungen wurden auch für das Fernsehen Standardisierungen und Normen angestrebt. Während der Zeit des analogen Farbfernsehens waren NTSC (**N**ational **T**elevision **S**ystems **C**ommittee; Nord- und Südamerika), PAL (**P**hase **I**nterlating **L**ine; Europa, Südamerika, Afrika und Asien) und SECAM (**S**équentiel **C**ouleur à **M**émoire; Frankreich, Afrika und UdSSR) die gängigsten Standards. Das grundlegende Verfahren zur analogen Übertragung von Videosignalen erfolgte bei diesen Standards nach VBS (**V**ideo **B**lanking and **S**ync) beziehungsweise deutsch BAS (**B**ild-**A**ustast-**S**ynchron). Die Auflösung und Bildwiederholrate bei VBS hängt von der zugrundeliegenden Fernsehnorm ab. Bei NTSC sind es 480 sichtbare Zeilen mit 640 sichtbaren Pixeln bei 59,94 Halbbildern pro Sekunde. Bei PAL und SECAM sind es 576 sichtbare Zeilen bei 720 sichtbaren Pixeln und 50 Halbbildern pro Sekunde (576i). Außerdem unterscheiden sich NTSC, PAL und SECAM auch bei der Modulation und der Art, wie Farbinformationen hinzugefügt werden. Im Zeitalter der digitalen Bildübertragung haben diese drei Übertragungsverfahren weitgehend an Bedeutung verloren. Sie sind uns aber in Form digitaler Bildformate für DVD-Video oder SDTV (**S**tandard **D**efinition **T**elevision) erhalten geblieben.

### Composite-Video mit CVBS oder FBAS

Die ersten normierten Fernsehsignale waren für die Übertragung von monochromen Bildern ausgelegt. In den USA und in Europa



Bild 2. Geöffneter Fernseher mit Kathodenstrahlröhre (Quelle: Shutterstock / Sergio Sergio).

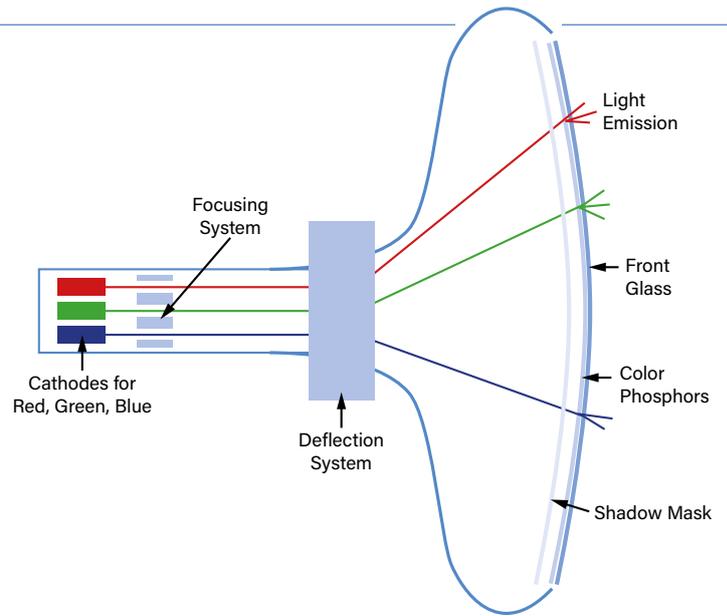


Bild 3. Schematischer Aufbau einer Farb-Bildröhre (Quelle: ITWissen.info).

wurden unterschiedliche Bildformate entwickelt. Das Timing eines VBS- oder BAS-Signals im PAL-Format von **Bild 1** dient als Beispiel für diese Standards. Dieser Standard eignete sich speziell für die Bildwiedergabe mit Kathodenstrahlröhren (**Bild 2**). Das Funktionsprinzip solcher Bildröhren unterscheidet sich massiv von LCDs oder OLED-Bildschirmen. Bei den Bildröhren wird ein modulierter und passend horizontal plus vertikal abgelenkter Elektronenstrahl auf eine Phosphorschicht gelenkt. Am Auftreffpunkt leuchtet der Phosphor dann proportional zur Intensität des Elektronenstrahls (**Bild 3**). Durch zeitlich synchronisierte horizontale und vertikale Ablenkung entsteht ein zweidimensionales Bild.

Aus Betrachterperspektive wird ein (schwarz-weißes oder besser: Graustufen-)Bild zeilenförmig von links nach rechts und die Zeilen dann von oben nach unten aufgebaut (**Bild 4**). Der erste „Bildpunkt“ pro Bild ist also links oben zu finden. Diese Logik hat sich auch bei den digitalen, modernen Displays weitgehend erhalten. Das historische Wissen um den Bildaufbau per wanderndem Elektronenstrahl erleichtert auch das Verständnis moderner Videosignale. Das Signal von Bild 1 enthält den Aufbau einer Bildzeile. Im vorderen Bereich (**Bild 5**) des Signals ist das „Horizontal blanking“ zu sehen. Zu Beginn des „Horizontal blanking“ befindet sich der Elektronenstrahl am rechten Rand der Bildröhre und muss daher für die

nächste Zeile wieder nach ganz links gebracht werden. Während dieses Sprungs von rechts nach links wird der Elektronenstrahl ausgestastet, in dem das Helligkeitssignal schwarz, ja sogar „dunkler als schwarz“ gemacht wird. Die horizontale Austastung (englisch „horizontal blanking“) dauert bei PAL 12  $\mu$ s und bei NTSC 10,9  $\mu$ s. Sie setzt sich aus einer sogenannten vorderen Schwarzschieler (englisch „front porch“), dem Zeilensynchronimpuls (englisch „sync tip“) und der sogenannten hinteren Schwarzschieler (englisch „black porch“) zusammen. Das Helligkeitsniveau des front porch (PAL = 1,65  $\mu$ s / NTSC = 1,4  $\mu$ s) liegt auf oder etwas unter dem Wert von schwarz bei 0,3 V, während der Elektronenstrahl noch weiter

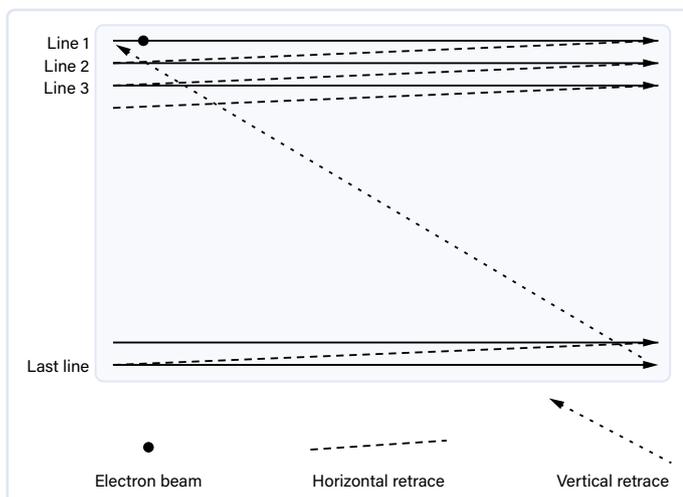


Bild 4. Zeilenförmiger Bildaufbau.

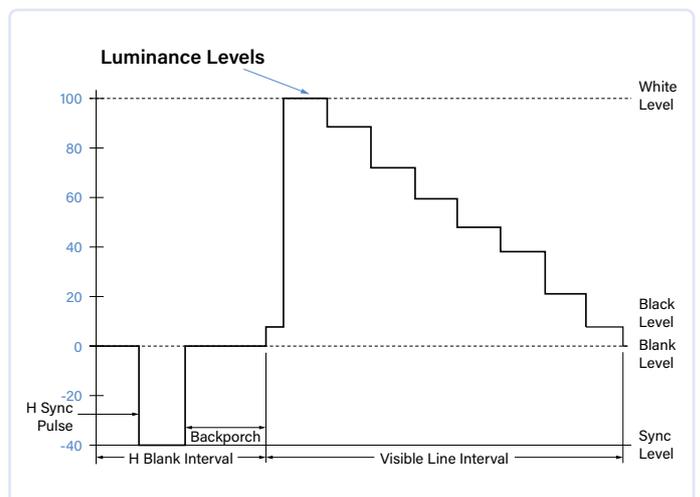


Bild 5. Horizontalaustastung (Quelle: www.edn.com).

nach rechts aus dem sichtbaren Bild hinausläuft. Die Spannung beim sync tip (PAL und NTSC = 4,7  $\mu$ s) liegt bei 0 V und ist damit deutlich tiefer als der Schwarzwert. Er sorgt dafür, dass der Elektronenstrahl von rechts nach links springt. Der black porch (PAL und NTSC = 4,7  $\mu$ s) gibt die Referenz für den Schwarzwert der Zeile von 0,3 V vor. Ab dem Ende des black porch fängt der zu zeichnende Inhalt, sprich, die Bildinformation an. Für den sichtbaren Teil einer Zeile stehen dann 52  $\mu$ s zur Verfügung. Die Anzahl der Pixel oder Abtastpunkte, die in diese 52  $\mu$ s gepackt werden, hängt vom Videoformat ab. Bei PAL sind es in der Regel 720 sichtbare Bildpunkte. Wenn alle Zeilen gezeichnet sind und der Elektronenstrahl rechts unten angekommen ist, muss ein Sprung zum Bildanfang durchgeführt werden. Dieser wird samt seiner Einleitung in einem eignen Abschnitt beschrieben.

### Synchronisation und Halbbilder

Die analoge Bandbreite für die Übertragung von Fernsehbildern reichte früher gerade für 25 Bilder pro Sekunde bei PAL beziehungsweise 29,97 Bilder pro Sekunde bei NTSC aus. Für das menschliche Auge ergibt sich damit zwar ein bewegtes Bild, allerdings ist es nicht sehr flüssig und erzeugt manchmal einen sehr unangenehm flimmernden Bildeindruck. Bei einer schnelleren Bildfolge von 50 oder gar 59,94 Bildern pro Sekunde nimmt das Flimmern spürbar ab. Die meisten Menschen erleben dies dann als flüssigen Bildeindruck. Da die zur Verfügung stehende Funkbandbreite begrenzt war, wurde die Halbbildtechnik eingeführt: Anstatt alle Bildzeilen höherfrequent als Vollbilder zu übertragen, werden abwechselnd die Zeilen mit geraden oder ungeraden Nummern mit der doppelten Frequenz von 50 Hz beziehungsweise 59,94 Hz übertragen (**Bild 6**). Zuerst wird also ein Halbbild mit allen ungeraden



Bild 6. Bildaufbau mit zwei Halbbildern.

den Zeilennummern übertragen und dann das nächste Halbbild mit geraden Zeilennummern. Damit die Zeilen der beiden Halbbilder gegeneinander in die Lücken des anderen geschrieben werden, gibt es eine Erkennung am Anfang jedes Halbbildes.

Da die Zeilen eines kompletten Bildes anhand der Übertragungsreihenfolge durchnummeriert sind, hat das erste Halbbild bei PAL die Zeilen 1 bis 313 und das zweite Halbbild zählt dann weiter bis Zeile 625. Die vertikale Synchronisation für das Ende eines Halbbildes und den Anfang des nächsten ist etwas verwickelt: Die Zeilen 311 bis 317 beim Übergang des ungeraden und geraden Halbbildes sowie die Zeilen 623 bis 5 beim Übergang des geraden zum ungeraden Halbbild des nächsten Komplet-

bildes sind nicht sichtbar. In diesen Zeilen werden schnellere Synchronisationsimpulse mit Pegeln zwischen 0,3 V und 0 V übertragen. **Bild 7** zeigt den Ablauf für ein komplettes PAL-Bild. Mit Hilfe dieser Synchronisationsimpulse kann die Elektronik erkennen, ob es sich um das Halbbild mit geraden oder ungeraden Zeilennummern handelt.

### Monochromes Video mit AVR-Controllern

Grundlegende Informationen zu Composite-Signalen wie CVBS oder FBAS finden sich bei Wikipedia [1]. Wer mehr Details und Schaltungstricks sucht, wird sicher im Buch „Analogue Video“ [2] von Angelo La Spina fündig werden.

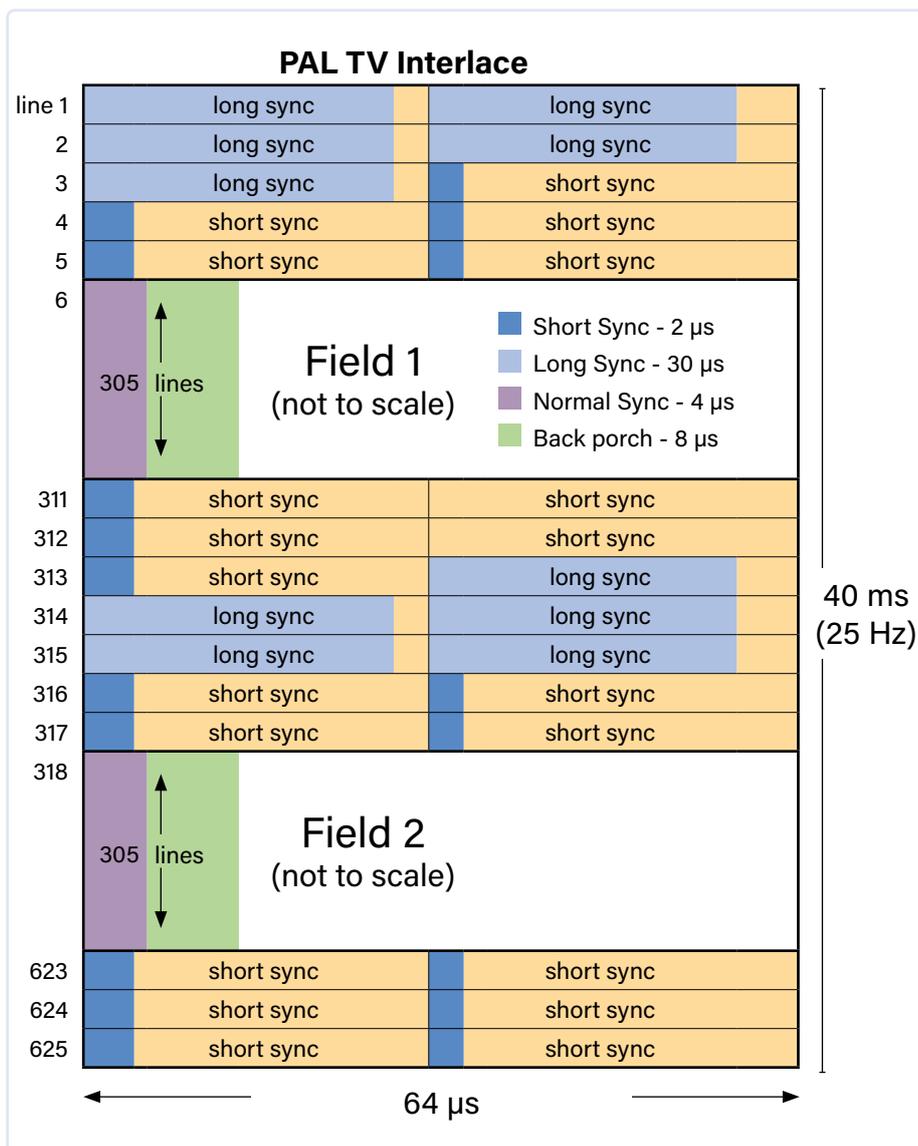


Bild 7. Bildaufbau eines kompletten PAL-Bildes (Quelle: martin.hinner.info).



Wird das Videosignal als VBS- oder BAS-Signal (keine Farbinformationen, sondern nur Helligkeit) mit Hilfe eines passenden Composite-Video-Kabels (**Bild 8**) übertragen, kann die Erzeugung der Helligkeitsinformationen recht einfach ausfallen. Wenn man statt Graustufen nur Weiß und Schwarz verwendet, muss man nur zwei passende Spannungen plus Synchronisation ausgeben. Neben der Vereinfachung der Schaltung verringert dies auch den für die Bilderzeugung nötigen Speicher.

Der Mikrocontroller Atmega328P eines Arduino Uno ist so in der Lage, ein monochromes Bild mit 128x96 Pixeln zu generieren. Alle Pixel können hierbei im internen Speicher des Arduino Uno vorgehalten werden, denn es werden nur 1.536 Byte dafür benötigt. Die Beschaltung eines Arduino Uno mit Composite-Ausgang wird in **Bild 9** gezeigt.

Wie man sieht, genügen zwei Widerstände, um ein passendes Signal ausgeben zu können. Für die Bildausgabe muss man das Rad nicht durch eigenen Code neu erfinden, sondern kann die sogenannte *TVOut*-Library [3] verwenden. Diese Bibliothek unterstützt das Timing für NTSC und PAL. Die damit machbare Videoausgabe reicht von simplem Text bis zum eigenen Spiel. Hackvision [4] ist eine Plattform für eine Spielkonsole. Dabei handelt es sich um Open-Hardware, für die im Extremfall ein Steckbrett und ein paar Bauteile genügen. Ein Arduino lässt sich ebenfalls zu einer Hackvision-Plattform (**Bild 10**) umrüsten und so die zugehörige Bibliothek an Spielen verwenden.

Die Videoausgabe mit einem AVR-Mikrocontroller benötigt einen seiner Timer. Die horizontale und vertikale Synchronisation wird durch Timer1 und dessen Ausgangspin PB1 (OC1A) bereitgestellt. Die Pixel werden timerbasiert passend Zeile für Zeile über einen eigenen Pin (PD7) ausgegeben.

Die Werte für die Widerstände an PB1 und PD7 lassen sich recht einfach ermitteln. Ein Composite-Eingang bei Monitoren oder TV-Geräten hat eine Eingangsimpedanz von 75 Ω. Die Pegel für die Synchronisation betragen 0 V und 0,3 V. Die Spannungen für die Helligkeitswerte der Pixel liegen bei 0,3 V für Schwarz und 1 V für Weiß. Im Folgenden wird davon ausgegangen, dass der Mikrocontroller mit 5 V versorgt wird.

**Bild 11** zeigt den Spannungsteiler für die Videoausgabe an der typischen Impedanz von 75 Ω. Pin 9 liefert die Synchronisationssignale.



Bild 8. Kabel für Composite-Video mit Cinch-Stecker (Quelle: Shutterstock / Woodpond).

Rechnerisch wäre für R1 ein Wert von 1175 Ω erforderlich, damit bei einem High-Pegel an Pin 9 über R2 0,3 V abfallen. 1 kΩ sorgt für maximal 0,34 V an R2 und ist genau genug für diese Zwecke. Pin 7 gibt die Helligkeitswerte aus. Benötigt werden nun Werte zwischen 0,34 V (Schwarz) und 1 V (Weiß) an R2. Um 1 V an R2 zu erzielen, ist rechnerisch für die Parallelschaltung von R1 und R3 ein Wert von 375 Ω notwendig, was für R3 dann 600 Ω ergäbe. Mit dem nächstgrößeren Wert von 470 Ω aus der E-12-Reihe ist sichergestellt, dass die Spannung an R2 den Wert von 1 V nie übersteigen kann.

Die Tatsache, dass das mit einem Arduino Uno oder einem Atmega328P funktioniert, beweist, dass eine nur geringe Rechenleistung und wenig Speicher ausreichen, um Grafiken auf einem Bildschirm auszugeben. Da das komplette Bild im internen Speicher vorgehalten werden kann, ist das Zeichnen eines neuen Bildes nicht zeitkritisch, lediglich die Erzeugung der Videosignale mit Hilfe von Timer1.

Lassen sich auch Graustufen = mehrere Helligkeitswerte mit dem Arduino ausgeben? Das geht, jedoch ist der interne Speicher der limitierende Faktor. Bei 16 Graustufen (4 Bit) und passenden Widerständen werden 6144 Byte für den Bildspeicher nötig, wenn man die Auflösung von 128x96 Pixel beibehalten möchte und ein komplettes Bild in den Speicher passen muss.

### Raspberry Pi Pico und Composite

Auch ein Raspberry Pi Pico kann ein Composite-Signal ausgeben, und das leicht mit mehr als 50 Graustufen. Dazu wird ein R2R-Netzwerk [5] als Digital/Analog-Konverter eingesetzt. **Bild 12** zeigt eine passende Schaltung, die aus Widerständen mit 180 Ω, 320 Ω und 360 Ω besteht. Wie der GitHub-Seite des

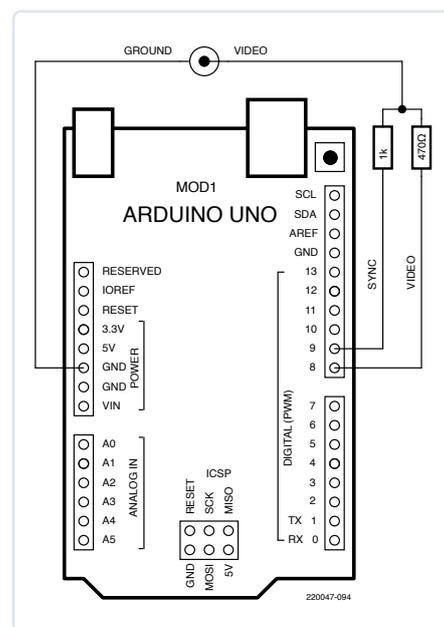


Bild 9. S/W-Videoausgang mit zwei Widerständen bei einem Arduino Uno.

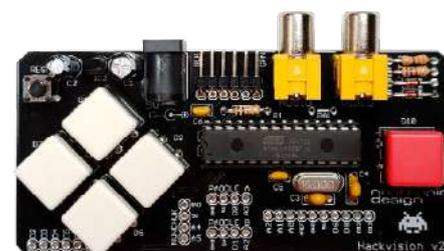


Bild 10. Hackvision Hardware (Quelle: nootropic design).

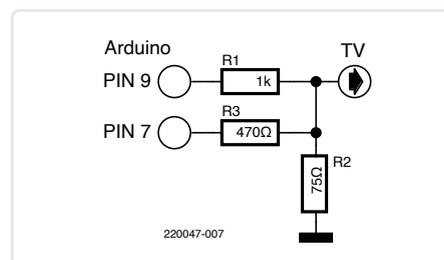


Bild 11. Die Schaltung des Videoausgangs von Bild 9.

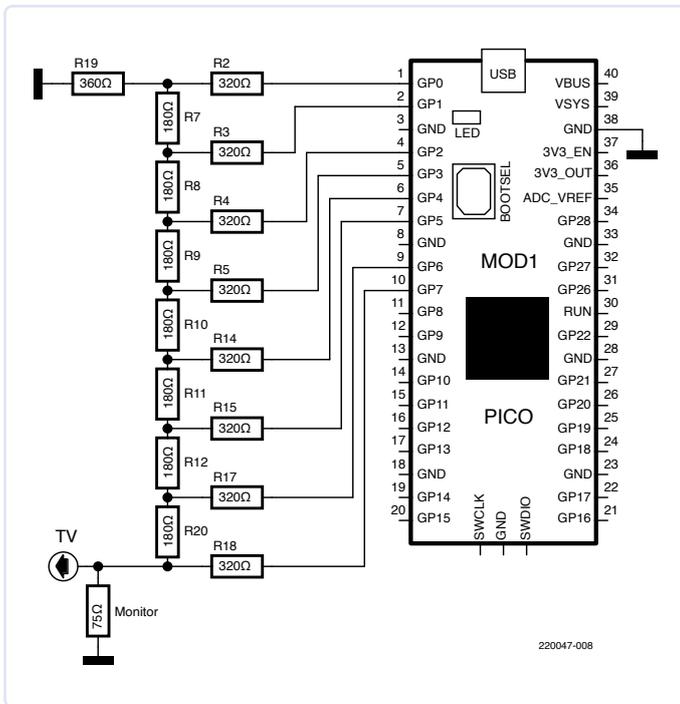


Bild 12. Ein R2R-Netzwerk als DAC beim Raspberry Pi Pico.

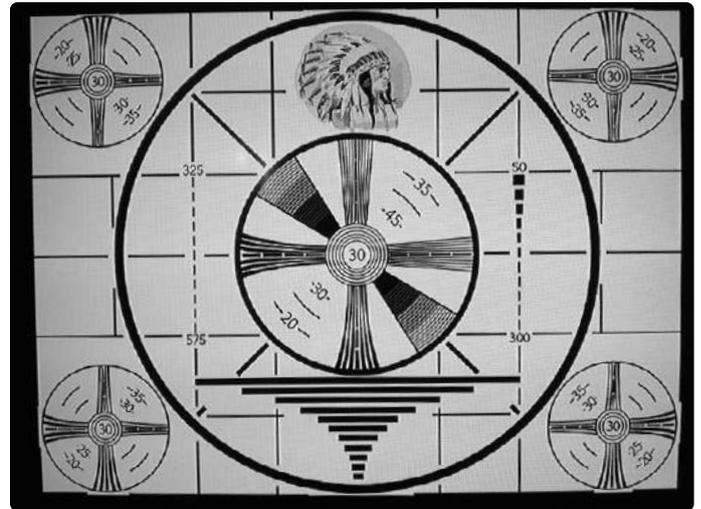


Bild 13. Das Graustufenbild eines Raspberry Pi mit 512x384 Pixel (Quelle: [tinyurl.com/2p8z27a2](https://tinyurl.com/2p8z27a2)).

Projekts „pico-composite8“ [6] entnommen werden kann, ist dabei der interne Widerstand der GPIO-Pins nicht zu vernachlässigen. Der Entwickler des Projektes hat hierfür Werte von etwa 40 Ω ermittelt. Der Raspberry Pi gibt hier ein Composite-Signal nach NTSC aus. Er lädt die Bilder entweder aus dem RAM oder aus dem Flash und kann maximal 512x384 Pixel ausgeben. Das Resultat ist in **Bild 13** zu bewundern.

Das Projekt ist allerdings lediglich eine Machbarkeits-Demonstration und bietet keine fertige generische Bibliothek. Dass ein Raspberry Pi Pico Video mit 512x384 Pixel in Graustufen ausgeben kann, zeigt, dass es hierfür weniger auf die Rechenleistung als auf das richtige Timing ankommt. Wenn ein komplettes Bild im RAM des Raspberry Pi Pico vorgehalten wird, stehen von den 264 kB nur noch etwa 64 kB für eigene Anwendungen zur Verfügung. Wenn man aber bedenkt, dass bei geschickter Programmierung selbst ein ATmega Spiele wie Tetris oder Pong samt Videoausgabe packt, sollte dieser Platz mehr als genug Raum für eigene Kreationen bieten.

### Farbe für Composite-Video

Mehr als 50 Graustufen sind gut, aber Farbe ist einfach besser. Wenn es um Composite-Video und Farbe geht, wird es deutlich schwieriger als bisher, das passende analoge Signal zu generieren. Rickard Gunée demonstrierte im Jahr 2003 auf einem Hackaday-Event [7] mit einem Scenix / Ubicom SX28 und etwa 50 MHz Takt die Genese eines Composite-Signals (PAL oder NTSC).

Doch wie schwierig ist es nun, ein Videosignal um Farbe zu erweitern? Die Antwort hängt von der Art und Weise ab, wie Farbinformationen für das Composite-Signal hinzugefügt werden.

### Farbig dank PAL und NTSC

Beim Umstieg von Schwarzweiß- auf Farbfernsehen vor einigen Jahrzehnten wurde kein gänzlich neues Signal definiert, den es musste sichergestellt sein, dass bestehende Schwarz-Weiß-Fernseher nach wie vor ein Bild anzeigen konnten. Dieses Problem wurde international auf drei verschiedene Arten gelöst, weshalb es zur Koexistenz der Standards NTSC, PAL und SECAM kam. Gemeinsam ist ihnen, dass die Farbinformationen zusätzlich in ein monochromes Signal integriert wurden.

Während sich PAL und NTSC prinzipiell ähneln (Quadraturmodulation für Farbe), unterscheidet sich SECAM bei der Einbettung der Farbe durch Verwendung von Frequenzmodulation.

Farben, die in ihren drei Grundkomponenten Rot, Grün und Blau vorliegen, müssen in einen YUV- oder YCbCr-Farbraum [9] umgerechnet werden, bevor diese dann passend über ein Composite-Videosignal ausgegeben werden können. Für die Umrechnung von RGB zu YUV gilt:

$$Y = 0,299 * R + 0,578 * G + 0,144 * B$$

$$U = 0,493 * (B - Y)$$

$$V = 0,877 * (R - Y)$$

Die Werte für R, G und B werden hier in einen Bereich zwischen 0 und 1 transformiert. Mit den Formeln ist ersichtlich, dass die Umrechnung für Mikrocontroller entweder Rechenleistung oder etwas Geschick beim Programmieren benötigt. Nachfolgend wird die Einbettung von Farbe anhand von PAL und NTSC in Grundzügen erläutert.

In Bild 1 ist ein monochromes PAL-Signal zu sehen, dem noch keine Farbinformationen beigemischt worden sind. Für Farbinformationen wird bei PAL ein Hilfsträger mit

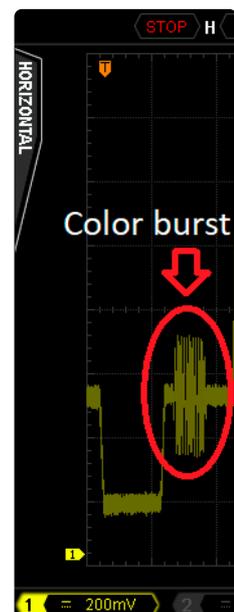


Bild 14. Oszillogramm des Color-Burst eines PAL-Signals.

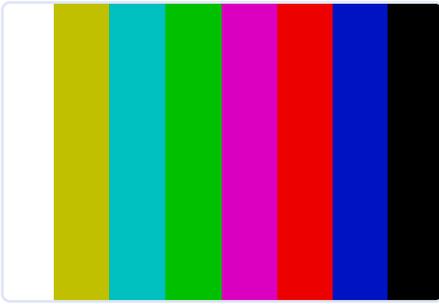


Bild 15. Testbild mit Farbbalken.

4,43361875 MHz verwendet, der als Referenzsignal genutzt wird. Diese Referenz wird in jeder Zeile als „Color burst“ (Bild 14) während der horizontalen Synchronisation gesendet. Mit Hilfe des Testbildes (Bild 15) wird erklärt, wie die unterschiedlichen Farben des Testbildes codiert sind. Das Testbild entspricht den EBU-Farbbalken [10]; ein Raspberry Pi Zero dient hier als Bildsignalgenerator.

Das Signal für eine Bildzeile ist in Bild 16 zu sehen. Helligkeit (weiße Linie als Mittelwert des Signals) und Amplitude der Farbinformation sind gut zu erkennen. In dem Signal steckt noch eine dritte Information, die in der Phasenlage verborgen ist. Bild 17 zeigt rot markiert die Phasenänderung beim Übergang in eine andere Farbe.

Damit sind im Signal die Informationen für die Helligkeit und die Amplitude des Farbsignals und die Phasenverschiebung vorhanden. Sollte es nun bei Übertragung oder Empfang des Signals zu einem Fehler in der Phasenlage oder dessen Auswertung kommen, so verändert sich der Farbeindruck des Bildes, was bei NTSC-Empfängern händisch durch eine Farbtonkorrektur [11] per Drehregler am Empfänger (später auch mit elektronischen Lösungen) ausgeglichen werden konnte. Bei PAL wurde dieses Problem vermieden, da die Phaseninformation mit jeder zweiten Zeile um 180° verschoben ist. Ein Fehler in der Phasenlage gleicht sich so zwischen zwei benachbarten Zeilen aus.

### Farbiges Composite-Video mit dem ESP32

Ein ESP32 kann mit ein paar kleinen Tricks leicht ein monochromes Composite-Signal ausgeben und selbst ein Arduino ist dazu in der Lage. Doch die Genese eines farbigen Composite-Signals stellt deutliche höhere Anforderungen an die Modulation.

Im Jahr 2018 demonstrierte bitluni seinen Ansatz der Erzeugung eines farbigen Composite-Signals mit einem ESP32 [12]. Es zeigte sich, dass die 13,33 MSa/s der DACs des ESP32 ausreichen, um einen Farbträger und Farbinformationen in das Signal zu integrieren. Wie schon beim monochromen Signal wird



Bild 16. Oszillogramm eines PAL-Farbsignals.



Bild 17. Die Phasenänderungen beim Farbwechsel sind markiert.

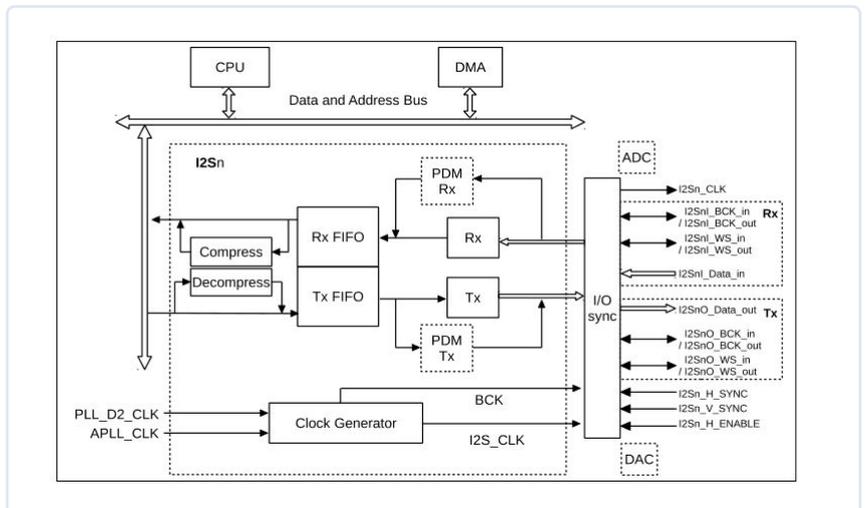


Bild 18. Der I2S-Block innerhalb des ESP32 (Quelle: Espressif / tinyurl.com/lyrrbnjak).

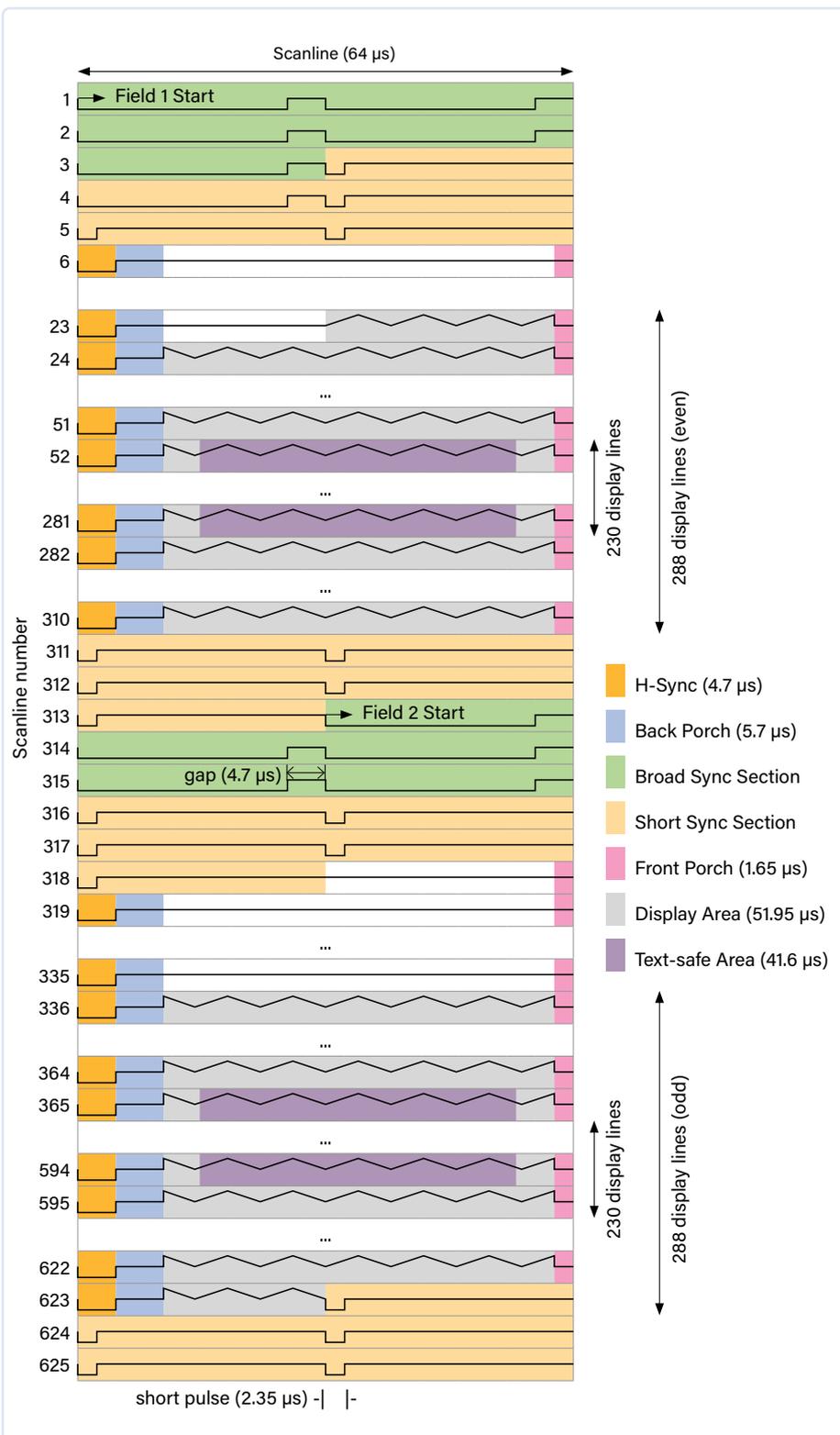


Bild 19. Signal der Bildausgabe mit halber Bildzeile (Quelle: batsocks.co.uk / <https://tinyurl.com/4fyhmk>).

ein Prozessorkern des ESP32 für die Genese des Videosignals verwendet. Das Setup nutzt den I2S-Block innerhalb des ESP32, um Daten an den DAC zu schicken (Bild 18). Bei PAL und bei NTSC wird das Bild interlaced, also nacheinander in Halbbildern mit ungeraden Zeilen (Odd-Field) und geraden Zeilen (Even-Field) ausgegeben. Wie schon

erwähnt, verdoppelt sich so die wahrgenommene Bildwiederholrate (bei gleicher Bandbreite), was das empfundene Flimmern deutlich reduziert. Das Timing dieser beiden Halbbilder hat ein paar Besonderheiten, wie etwa nur eine halbe Bildzeile am Ende des Bildes in einem der beiden Halbbilder (Bild 19).

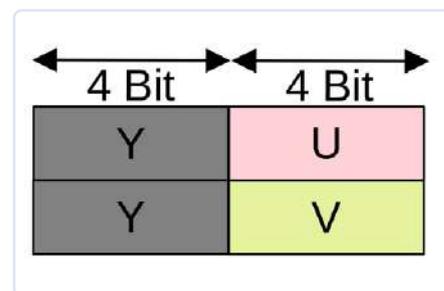


Bild 20. Speicherung der Farbinformation bei bitluni.

Da das RAM eines ESP32 nur dazu ausreicht, ein Bild mit halber Auflösung auszugeben, müssen Zeilen doppelt gezeichnet werden: Der Inhalt von Zeile 1 erscheint demnach auch in Zeile 2. Das Interlacing sorgt so dafür, dass zweimal der gleiche Inhalt gezeichnet wird. Nun stellt sich die Frage, ob man wirklich Odd-Field und Even-Field ausgeben soll, oder ob es auch reichen würde, zweimal das Even-Field zu nutzen und damit den Code für die Ausgabe deutlich zu vereinfachen. Genau dieser Trick wurde schon vom SNES (Super Nintendo Entertainment System) angewendet. Und auch der Code von bitluni nutzt dieses Verfahren, um den Code für die Synchronisierung zu vereinfachen. Es werden also lediglich 288 echte Zeilen bei 50 Hz ausgegeben (288p) – für NTSC wäre der äquivalente Modus 240 Zeilen bei 60 Hz (240p).

Gegenüber der monochromen Bildausgabe müssen nun auch Farbinformationen im RAM des ESP32 vorgehalten werden. In der Regel werden die Farben durch ihre Komponenten Rot, Grün und Blau repräsentiert. Für die Erzeugung eines normgerechten Composite-Signals sollten die Farbinformationen aber besser als YUV-Werte im RAM abgelegt sein. Nur so kann der ESP32 und sein DAC diese Daten schnell genug ausgeben. Da die Menge an RAM im ESP32 limitiert ist, muss bei der Codierung der Daten etwas getrickst werden. bitluni legt die Infos als kombinierte YU-, YV- und V-Werte mit jeweils 4 bit Auflösung im RAM ab (Bild 20).

Beim Ansatz von bitluni ist leider nur die Ausgabe als Composite-Video nach PAL-Norm möglich, nach NTSC klappt es nicht. Die 3,57954 MHz des NTSC-Color-Burst haben so ein ungünstiges Verhältnis zur Samplerate des DAC, dass kein verwertbarer Color-Burst ausgegeben wird; Empfänger können sich nicht synchronisieren.



Bild 21. Emulatoren für 8-bit-Spielkonsolen (Quelle: [tinyurl.com/ykd9ezap](https://tinyurl.com/ykd9ezap)).

einige Zugeständnisse an das Videomaterial nötig. Details dazu lassen sich auf der zugehörigen Github-Seite [14] finden.

### Demnächst im zweiten Teil

In zweiten Teil des Artikels geht es um VGA, DVI und Sprites. Vor allem mit einem ESP32 und dem RP2040 des Raspberry Pi Pico sind hier erstaunliche Effekte möglich. Wer nicht bis zum zweiten Teil warten mag, kann sich für das Webinar „Microcontroller as Pixel Artist“ [15] registrieren. Dort wird es dann ebenfalls um die Themen VGA, DVI und Sprites gehen. ◀

220047-02



### Webinar „Microcontroller as Pixel Artist“

Da bewegte Bilder gerade bei diesem Thema mehr sagen als Worte und auch umfangreiche Listings nicht mehr in den Artikel gepasst haben: Freuen Sie sich auf das Webinar zu dieser Ausgabe. Unter [15] können Sie sich schon jetzt registrieren. Es wird um Grafik mit Arduino, ESP32 und Raspberry Pi Pico gehen.

### Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [mathias.claussen@elektor.com](mailto:mathias.claussen@elektor.com) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

### Farbe mit ESP32 in PAL und NTSC

Der GitHub-User „rossumur“ hat mit seinem Projekt „esp\_8\_bit“ bewiesen, dass sich auch Farbe nach NTSC mit einem ESP32 erzeugen lässt. Bei diesem Projekt handelt es sich um eine Sammlung von ESP32-Emulatoren für diverse 8-Bit-Konsolen, vom Atari 400 über das Nintendo Entertainment System bis hin zum Sega Master System (Bild 21).

Es wird einfach ein Pin des ESP32 genutzt, um ein Composite-Signal in NTSC oder PAL auszugeben. Der Trick liegt in der Audio-PLL des ESP32 verborgen. Diese kann dazu genutzt werden, um den DAC mit Sampleraten bis zu etwa 20 MHz zu betreiben. Die vierfache Frequenz des NTSC-Farbträgers liegt bei 14,318182 MHz (bei PAL 17,734475 MHz). Mit der APLL lassen sich 14,318180 MHz und 17,734476 MHz erzeugen, was nahe genug an den nötigen Frequenzen für PAL und NTSC liegt. Damit gibt der DAC

nun ein ganzzahliges Vielfaches der Farbträgerfrequenz aus, was zu verarbeitbaren Videosignalen führt.

Mit der APLL kann der DAC des ESP32 nicht nur ein Videosignal ausgeben, dieser Ansatz ist auch für andere Anwendungen aus dem Bereich DDS (Direct Digital Synthesis) sehr interessant. Die Verwendung der APLL hat bei hohen Sampleraten jedoch einen Nachteil. Der DAC des ESP32 hat zwei Kanäle. Wird nun einer von der APLL mit Daten für die Videoausgabe versorgt und versucht man, den zweiten Kanal mit Audiodaten aus dem I<sup>2</sup>S-Interface zu beliefern, kommt es zu Aussetzern des DACs. Diese Aussetzer zeigen sich auf beiden Kanälen, so dass man für die Ausgabe von Audio auf andere Methoden zurückgreifen muss.

Auf Basis des Projektes „esp\_8\_bit“ aufbauend ist „espflix“ entstanden. Der ESP32 kann so Videos wiedergeben, die auf einem Server des espflix-Projektes liegen. Dazu sind aber

### WEBLINKS

- [1] Composite Video (Wikipedia): [https://de.wikipedia.org/wiki/Composite\\_Video](https://de.wikipedia.org/wiki/Composite_Video)
- [2] Angelo La Spina, „Analogue Video“: <https://www.elektor.com/analogue-video-e-book>
- [3] Arduino TVOut Library: <https://github.com/Avamander/arduino-tvout>
- [4] Hackvision: <https://nootropicdesign.com/hackvision/>
- [5] R2R-Netzwerk (Wikipedia): <https://de.wikipedia.org/wiki/R2R-Netzwerk>
- [6] pico-composite8: <https://github.com/obstruse/pico-composite8>
- [7] Hackaday-Event: <https://hackaday.com/2022/08/17/chips-remembered-the-scenix-ubicom-parallax-sx>
- [8] Farbe mit SX-Chips: <https://elinux.org/images/e/eb/Howtocolour.pdf>
- [9] YCbCr-Farbraum (Wikipedia): <https://de.wikipedia.org/wiki/YCbCr-Farbmodell>
- [10] EBU-Farb-Balken: <https://en.wikipedia.org/wiki/YUV>
- [11] NTSC-Farbkorrektur (Wikipedia): [https://en.wikipedia.org/wiki/Tint\\_control](https://en.wikipedia.org/wiki/Tint_control)
- [12] bitluni, „ESP32 Composite Video“: <https://bitluni.net/esp32-composite-video>
- [13] Github-User rossumur: <https://github.com/rossumur>
- [14] espflix: <https://github.com/rossumur/espflix>
- [15] Webinar „Microcontroller as Pixel Artist“: <https://www.elektormagazine.com/webinars>



### Passende Produkte

- **Raspberry Pi Pico RP2040 (SKU 19562)**  
[www.elektor.de/19562](http://www.elektor.de/19562)
- **ESP32-DevKitC-32D (SKU 18701)**  
[www.elektor.de/18701](http://www.elektor.de/18701)
- **Arduino Uno Rev3 (SKU 15877)**  
[www.elektor.de/15877](http://www.elektor.de/15877)



# electronica 2022

## Neues von der Weltleitmesse

Von Stuart Cording und Jens Nickel

Das nennt man wohl einen vollen Erfolg: Auf die electronica 2022 vergangenen November strömten rund 70.000 Besucher. Mitunter konnte es da ganz schön eng werden - zumindest auf den Ständen der großen Distris und Halbleiterhersteller. Und die haben bei der weltgrößten Messe ihrer Art nicht nur gekleckert, sondern geklotzt, mit opulenten Ständen und attraktiven Exponaten. So gab es Motorräder mit eingebauter KI zu bestaunen, Spezial-Vans zur Drohnerdetektion, jede Menge Rennwagen und auch einen 1-Personen-Quadropter-Simulator. Eine kleine Auswahl interessanter Dinge zeigen wir hier.



Aeronia, Spezialist für (HF-)Messtechnik, zeigte diesen Van zur Drohnerdetektion. Der vollausgestattete Wagen ist für rund 1,5 Millionen Euro erhältlich, doch werden auch weit günstigere Lösungen zum Aufspüren der unbemannten Flugkörper angeboten, etwa in Racks.

[www.aeronia.de/drohnen-detektor](http://www.aeronia.de/drohnen-detektor)



Digitale Mikroskope verdrängen mehr und mehr die Vertreter mit Okular. Sie sind bedienfreundlich auch für mikroskopie-ungeübtes Prüfpersonal und erlauben eine automatische Durchmusterung von Proben. Das hier gezeigte Exemplar von Olympus fertigt sogar 3D-Bilder an, die aus allen Perspektiven betrachtbar sind.

[www.olympus-ims.com/de/microscope/dsx1000/high-end-model](http://www.olympus-ims.com/de/microscope/dsx1000/high-end-model)



Während die virtuelle Realität darum kämpft, sich durchzusetzen, bietet Augmented Reality (AR) greifbare Vorteile für die Industrie. Techniker können bei der Reparatur einer Maschine Zugang zu Anweisungen oder Anleitungen erhalten, die direkt vor ihre Augen gebeamt werden, während sie gleichzeitig ihre Umgebung vollständig im Blick haben. Das Farb-Lasermodule von TDK, das in einem Prototyp einer AR-Brille integriert ist, bildet Farbbilder direkt auf die Netzhaut ab - und wir haben es überlebt, um unsere Erfahrungen mit Ihnen zu teilen!

[www.tdk.com/de/news\\_center/press/20221013\\_01.html](http://www.tdk.com/de/news_center/press/20221013_01.html)





Sicher das Highlight am Stand des Halbleiterherstellers Infineon: Wer etwas Geduld mitbrachte, konnte einen virtuellen Flug mit diesem 1-Personen-Quadroptersimulatorwagen.  
[www.infineon.com/cms/en/product/promopages/electronica/](http://www.infineon.com/cms/en/product/promopages/electronica/)



E-Paper-Displays sind auf dem Vormarsch, denn sie bieten ein knackscharfes, blickwinkelstabiles Bild. Distributor Beck Elektronik zeigte diese mehr als DIN-A4-großen Exemplare in monochrom und Farbe. Im Vergleich zu LCDs sind die Displays aber (noch) teuer. Überdies leuchten sie auch nicht selbstständig, halt genauso wie bedrucktes Papier.  
[www.beck-elektronik.de/produkte/displays/e-paper-display-epd](http://www.beck-elektronik.de/produkte/displays/e-paper-display-epd)



Die Automobilindustrie verlässt sich zunehmend auf Software, um die Fähigkeiten ihrer Fahrzeuge mit Hilfe standardisierter, programmierbarer Hardware zu definieren. NXP hat den S32K39 mit zwei speziellen Motorcontroller-Peripheriegeräten vorgestellt. Sie unterstützen Steuerkreise mit bis zu 100 kHz und lassen sich mit breitbandigen Halbleitern aus Siliziumkarbid kombinieren, um hocheffiziente Antriebsrichter für Elektrofahrzeuge zu liefern.  
[www.nxp.com/products/processors-and-microcontrollers/s32-automotive-platform/s32k-general-purpose-mcus:S32K-MCUS](http://www.nxp.com/products/processors-and-microcontrollers/s32-automotive-platform/s32k-general-purpose-mcus:S32K-MCUS)



Die Angst vor der Reichweite ist eine der Hauptsorgen der Verbraucher, wenn sie von Autos mit fossilen Brennstoffen auf Elektroautos umsteigen. In der Hoffnung, diese Befürchtungen zu zerstreuen, hat Mercedes-Benz kürzlich bekannt gegeben, dass der VISION EQXX eine Strecke von 1.202 km (747 Meilen) mit einer einzigen Ladung zurückgelegt hat. Das Fahrzeug, das auf dem Stand von onsemi stand, nutzt deren VE-Trac-SiC-Module im Antriebsstrang, um 95% der Energie aus der Batterie an die Räder zu bringen.  
<https://elektor.link/onsemiSiC>

Die Elektor-Redaktion führte auf der electronica 2022 mehrere Interviews mit Ingenieuren und Vordenkern der Branche. Alle Interview-Videos finden Sie auf unserem YouTube-Kanal:  
[www.elektor.tv](http://www.elektor.tv)

( 220652-02)RG



OKdo ist eine Tochterfirma von RS Components, die Boards und Kits für Maker und Professionals anbietet. Die Single-Board-Computer der Marke ROCK sind gut ausgestattet, leistungsfähig und (weitgehend) lieferbar. Der neue Rock-5-Model B besitzt einen 8K-Videoausgang, was man mit einer Überwachungs-Applikation demonstrierte.  
[www.okdo.com](http://www.okdo.com)  
<https://wiki.radxa.com/Rock5>



# ESP32-Kamera

So einfach, dass sie nicht einmal WLAN braucht



Von Bera Somnath (Indien)

Brauchen Sie für eine Anwendung eine Kamera? Bevor Sie eine neue Kamera von der Stange kaufen, sollten Sie einen Do-it-yourself-Ansatz in Betracht ziehen. Mit einem ESP32-Board mit Kamera und ein paar zusätzlichen Komponenten können Sie eine einfache, aber effektive kundenspezifische Lösung bauen.



Alle Handys können Fotos machen, speichern und anzeigen. Daher ist es für niemanden eine ernstzunehmende Aufgabe, eine Kamera zu bauen, schon gar nicht für mich. Es lohnt sich höchstens, wenn sie wirklich sehr einfach zu realisieren ist. Mit ein paar preiswerten Modulen wie einem ESP32-Board mit Kamera, einem kleinen OLED-Display, einer Echtzeituhr und einem Auslösemechanismus können Sie eine batteriebetriebene Kamera bauen, die vollständig unter Ihrer Kontrolle Fotos knipst. Auf Druck auf den Auslöser oder durch Anwesenheitserkennung mit einem Bewegungssensor oder sogar durch einen berührungslosen Temperatursensor nimmt sie (diskret) das Bild einer Person auf, deren Körpertemperatur zu hoch ist. Der Bau einer solchen Anwendung ist sinnvoll, weil Sie Ihr Smartphone auf diese Weise nicht einsetzen können.

## ESP32-Cam

Ein PIR-Sensor rüstet ein ESP32-Cam-Board so um, dass es heimlich Portraits von missgünstigen Eindringlingen macht! Oder Sie verwenden nur den ESP32 mit Kamera und OLED-Display, um eine nette kleine Webcam zu bauen, die sich mit jedem verfügbaren Netzwerk in der Umgebung verbindet und den Videostream im Intranet veröffentlicht. Wenn Sie die Port-Forwarding-Funktion Ihres Modems oder Routers nutzen, können Sie das Bild sogar

im Internet veröffentlichen. Diese Funktion ist sehr praktisch für jemanden, der oder die eine zweite Kamera für Demozwecke in einem Webinar oder Online-Kurs benötigt. Die Kosten für das ESP32-Cam-Modul, das wir in diesem Artikel verwenden wollen, liegen bei etwa 10 €.

## Zunächst eine coole Demo

Nach der Installation der neusten Arduino-IDE und des neusten ESP32-Board-Pakets über die Board-Verwaltung der IDE wählen Sie als Board das „AI Thinker ESP32-CAM“ (*Werkzeuge* → *Board* → *ESP32 Arduino*) und den Port, an den es angeschlossen ist (*Werkzeuge* → *Port*). Öffnen Sie dann das ESP32-Webcam-Beispielprojekt (*Datei* → *Beispiele* → *ESP32* → *Camera* → *CameraWebserver*). Probieren Sie es aus, Sie werden überrascht sein. Die beiden folgenden Projekte sind Erweiterungen dieses Projekts. Es kann losgehen!

## Projekt 1: Webcam mit OLED-Display

Dieses Webcam-Projekt erlaubt es nicht, ein WLAN zum Beispiel aus einem Menü auszuwählen, weil die Netzwerk-Zugangsdaten fest im Code programmiert sind. Um aber die IP-Adresse der Webcam zu ermitteln, muss sie an einen PC mit einem seriellen Terminal angeschlossen werden. Hier erweitern wir das Projekt so, dass es

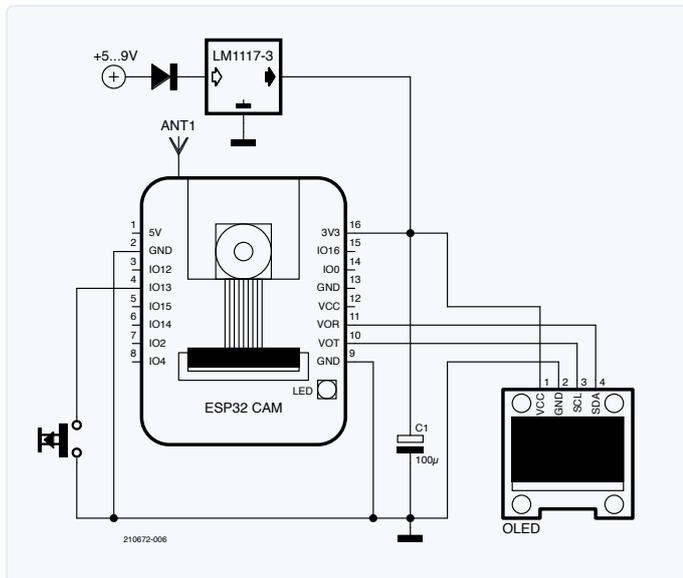


Bild 1. Fügen Sie der ESP32-Cam ein OLED-Display für zusätzlichen Komfort hinzu.

eine Verbindung zu jedem in der Nähe befindlichen Netzwerk herstellen kann. Sobald sie verbunden ist, zeigt sie auf dem kleinen OLED, mit welchem Netzwerk sie verbunden wurde und wie ihre IP-Adresse lautet. Wenn die Verbindung aus irgendeinem Grund unterbrochen wird, versucht sie, sich erneut mit einem der bekannten Netzwerke zu verbinden.

### Aufbau der Webcam

**Bild 1** zeigt, wie man das OLED-Display an das ESP32-Cam-Board anschließt. Neben dem Anschluss der SD-Karte und der Kamera sind nur noch sehr wenige GPIO-Pins für uns verfügbar: IO1, IO3, IO4, IO12 und IO13 (siehe Kasten **Elektor Labs Notes**). IO1 (UoT) und IO3 (UoR) werden auch als serielle Schnittstelle zum Hochladen des Programms auf den ESP32 im Flash-Programmiermodus verwendet. Daher müssen diese Pins beim Hochladen von Programmen völlig frei von jeglicher Beschaltung sein.

In diesem Webcam-Projekt können wir die für die SD-Karten-Anschlüsse reservierten Pins problemlos verwenden, da wir die SD-Karte nicht nutzen. Allerdings haben wir IO1 und IO3 als I<sup>2</sup>C-Pins für den Anschluss des OLED-Displays verwendet. Auf diese Weise haben wir die Verfügbarkeit unserer freien GPIO-Pins maximiert. Vergessen Sie aber nie, den I<sup>2</sup>C-Bus zu unterbrechen, wenn Sie ein Programm auf den ESP32 hochladen (**Bild 2**).

### Verwendung

Das Funktionsprinzip der ESP32-Webcam mit OLED-Display ist sehr einfach geblieben. Der ESP32 sucht nach den WLANs, die im Programm aufgelistet sind, und versucht, sich mit einem von ihnen zu verbinden. Sobald die Verbindung erfolgreich ist, erscheint auf dem OLED-Display der Status „Camera ready“ zusammen mit der Netzwerk-ID und der IP-Adresse, die der Webcam zugewiesen wurde. Um auf die Kamera zuzugreifen, verbinden Sie sich mit dem angezeigten Netzwerk und geben in einem Browser die IP-Adresse der Webcam ein. Die Portadresse wird standardmäßig beibehalten (8080).

Die Software für die ESP32-Webcam mit OLED-Display kann von

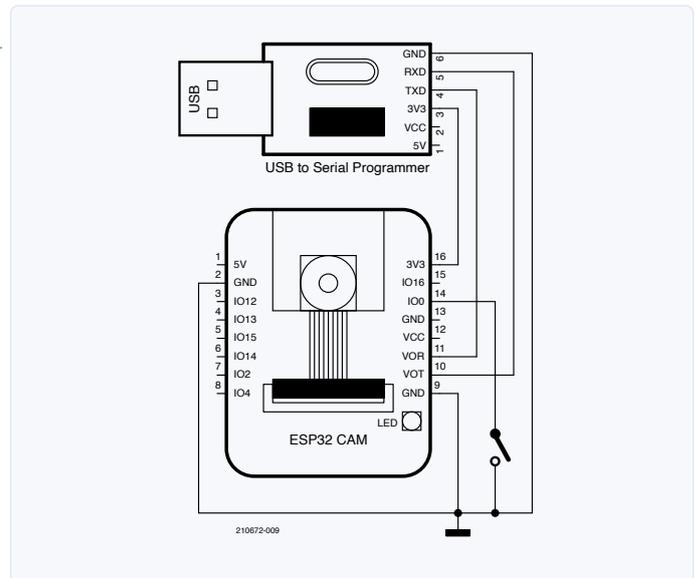


Bild 2. Die ESP32-Cam verfügt über keine integrierte USB-zu-Seriell-Schnittstelle. Um den Sketch hochzuladen, benötigt man daher einen USB-zu-Seriell-Adapter. Um in den Programm-Upload-Modus zu gelangen, muss der Reset-Knopf des ESP32 gedrückt werden (oder die Versorgung aus- und eingeschaltet werden), während IO0 auf Masse gezogen ist.

[1] heruntergeladen werden. Beachten Sie, dass sie die Bibliothek „ESP8266 and ESP32 OLED driver for SSD1306 displays“ (wir haben Version 4.1.0 verwendet) [2] benötigen. Sie kann mit dem Bibliotheksverwalter der Arduino-IDE installiert werden. Vergessen Sie nicht, den I<sup>2</sup>C-Bus zu trennen, bevor Sie einen Sketch hochladen!

### Projekt 2: Getriggerte Kamera mit OLED und RTC

Bei diesem Projekt wird jedes Mal ein Bild aufgenommen, wenn ein Ereignis eintritt. Die Bilder werden mit einer Seriennummer und einem Zeitstempel auf einer SD-Karte gespeichert.

Die Schaltung in **Bild 3** ist eine Erweiterung der vorherigen Schaltung. Ein Echtzeituhr-Modul (RTC) DS3231 wird an den I<sup>2</sup>C-Bus und eine Triggerquelle an IO13 angeschlossen.

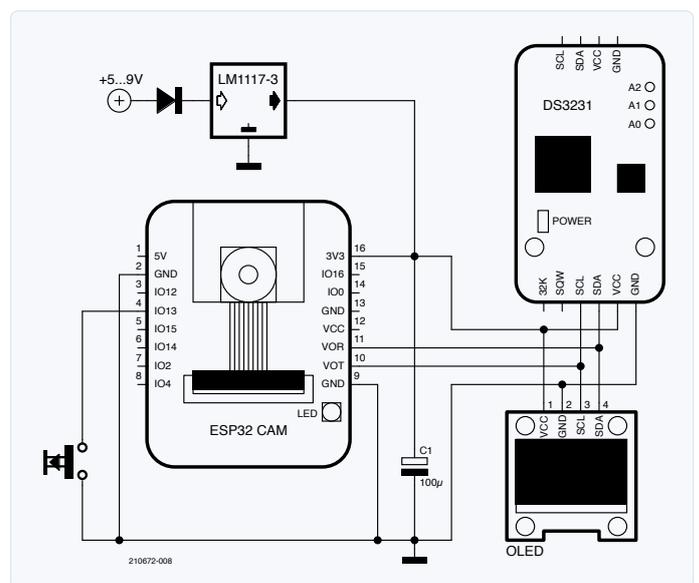


Bild 3. Ein Taster und ein RTC-Modul verwandeln die Webcam aus Bild 1 in eine zeitgesteuerte Überwachungskamera.

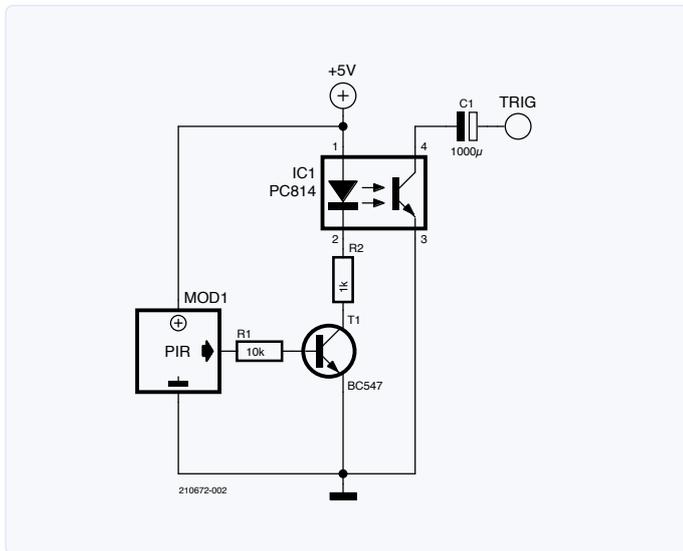


Bild 4. Wir können einen 5-V-PIR-Sensor verwenden - einige funktionieren sogar schon mit 3,3 V - um die ESP32-Kamera auszulösen. Um eine vollständige Isolierung zu erreichen, empfehlen wir die Verwendung eines solchen Optokopplers (oder eines Relais).

Dies kann ein Taster oder ein PIR-Sensor (**Bild 4**) oder etwas anderes sein, das einen kurzen, low-aktiven Impuls erzeugt, wenn ein Bild aufgenommen werden soll.

IO4 wird als Blitzlicht verwendet, wobei die superhelle weiße LED der ESP32-Cam als zusätzliche Lichtquelle für die Kamera eingesetzt wird. IO13 wird wie gesagt als Auslöser der Kamera eingesetzt. Das Programm schreibt dank der RTC das Datum und die Uhrzeit in den Dateinamen und speichert das Bild auf der SD-Karte. Anschließend geht die Kamera in den Deep-Sleep-Modus über, um die Batterie zu schonen, wenn keine Bilder aufgenommen werden. Ein low-aktiver Impuls an IO13 weckt dann die Kamera auf, um erneut ein Bild aufzunehmen.

Dieses Programm besteht nur aus der Funktion `setup`, die Funktion `loop` wird nicht verwendet. Der Grund dafür ist der Gebrauch des Tiefschlafmodus. Wenn das Gerät aus diesem Modus aufwacht, startet das Programm neu, führt die Funktion `setup` aus und nimmt ein Bild auf, bevor es wieder in den Tiefschlafmodus übergeht, so dass eine Funktion `loop` nie erreicht wird.

Die Software für die getriggerte Kamera kann von [1] heruntergeladen werden. Neben der OLED-Treiberbibliothek, die auch für das erste Projekt benötigt wird [2], ist hier auch die Bibliothek „Rtc by Makuna“ erforderlich. Wir haben von der Downloadquelle [3] die Version 2.3.5 verwendet. Auch diese Bibliothek kann mit dem Bibliotheks-Verwalter der IDE installiert werden. Und: Vergessen Sie nicht, den I<sup>2</sup>C-Bus abzukoppeln, bevor Sie einen Sketch hochladen (Bild 2)!

## Fazit

Die ESP32-Cam ist sehr billig, bietet aber trotzdem viele Möglichkeiten. Durch die Installation mehrerer dieser Kameras in einer Reihe können sie als fester und weitreichender

Objektscanner (50...100 m) verwendet werden, um Bilder von sich nähernden Fahrzeugkennzeichen zu machen (Geschwindigkeitsüberschreitung). Im Solomodus mit einem PIR-Sensor kann sie als „Wildcam“ für Vögel oder andere Tiere eingesetzt werden. Es sind aber auch komplexere Anwendungen denkbar. Durch Hinzufügen einer Gesichtserkennung und eines berührungslosen Temperatursensors kann die ESP32-Cam beispielsweise eine gewisse Sicherheit gegen COVID bieten. ◀

(210672-02)RG

## Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



## STÜCKLISTE

- ESP32-Cam Platine
- SD-Karte
- SSD1306-kompatibles OLED-Display (0,96" I2C)
- I2C-RTC-Modul DS3231 oder DS1307
- USB-zu-Seriell-Adapter (Kabel)
- 3,3 V Regler LM1117
- PIR-Sensor oder Taster
- Optional: Batterie, Kondensatoren, etc.



## PASSENDE PRODUKTE

- ▶ **ESP32-Cam-CH340 Development Board (SKU 19333)**  
[www.elektor.de/19333](http://www.elektor.de/19333)
- ▶ **0.96" I<sup>2</sup>C-OLED-Display (SKU 18747)**  
[www.elektor.de/18747](http://www.elektor.de/18747)
- ▶ **SparkFun Real Time Clock Module - RV-8803 (Qwiic) (SKU 19646)**  
[www.elektor.de/sparkfun-real-time-clock-module-rv-8803-qwiic](http://www.elektor.de/sparkfun-real-time-clock-module-rv-8803-qwiic)

## WEBLINKS

- [1] Download für dieses Projekt bei Elektor Labs: <https://www.elektormagazine.de/labs/esp32-camera>
- [2] ESP8266- und ESP32-OLED-Treiber für SSD1306-Displays: <https://github.com/ThingPulse/esp8266-oled-ssd1306>
- [3] Rtc by Makuna: <https://github.com/Makuna/Rtc>
- [4] Getriggerte Kamera, Version Elektor Labs: <https://www.elektormagazine.de/210672-01>

## Elektor Lab Notes

Viele ESP32-Cam-Boards werden zusammen mit einem praktischen USB-zu-Seriell-Wandler-Tochterboard\* verkauft (vielleicht sollten wir es Mutterboard nennen, da das ESP32-Modul darauf gesteckt wird und auch die Stromversorgung davon erhält). Diese Kombination (**Bild 5**) macht das Programmieren des ESP32 sehr einfach, aber sie kollidiert mit dem I<sup>2</sup>C-Port an den Pin IO1 und Pin IO3, die der Autor in seinem Projekt nutzt. Wenn man diese Pins benutzt, muss der I<sup>2</sup>C-Port leider jedes Mal abgetrennt werden, wenn man den ESP32 neu programmieren will, und deshalb haben wir nach einer anderen Möglichkeit für den I<sup>2</sup>C-Bus gesucht und gefunden.

Theoretisch kann der I<sup>2</sup>C-Bus auf dem ESP32 so konfiguriert werden, dass fast jeder seiner IO-Pins verwendet werden kann, aber auf dem ESP32-Cam-Board führt dies zu allen Arten von Boot-Problemen und PSRAM-Speicherfehlern. Das liegt an den Pull-Up-Widerständen des I<sup>2</sup>C-Busses.

Das ESP32-Cam Board besitzt zwei 8-fache Erweiterungsverbinder. Auf den ersten Blick scheint das in Ordnung zu sein, aber wenn man genauer hinsieht, sind die Dinge nicht so einfach. Zunächst einmal werden sechs der sechzehn Pins für die Stromversorgung verwendet, so dass nur zehn für IO-Ports übrig bleiben. Sechs dieser zehn (IO2, IO4, IO12, IO13, IO14 und IO15) werden mit dem SD-Kartensteckplatz geteilt. Damit bleiben nur die Ports IO0, IO1, IO3 und IO16 übrig. Das Hochladen eines Programms auf den ESP32 mit angeschlossenem Tochterboard benötigt die Ports IO0, IO1 und IO3. IO16 wird mit dem Chip-Select-Pin des PSRAM-Chips geteilt, so dass er nur sicher verwendet werden kann, wenn PSRAM nicht benötigt wird.

Die Bibliothek *SD card* ist standardmäßig auf den 4-Bit-Datenmodus eingestellt, kann aber in den 1-Bit-Datenmodus versetzt werden mit

```
SD_MMC.begin("/sdcard", true);
```

Und damit sind die Ports IO4, IO12 und IO13 freigegeben. Port 4 wird mit der hellen weißen Blitz-LED geteilt. Er hat einen Pull-up-Widerstand von

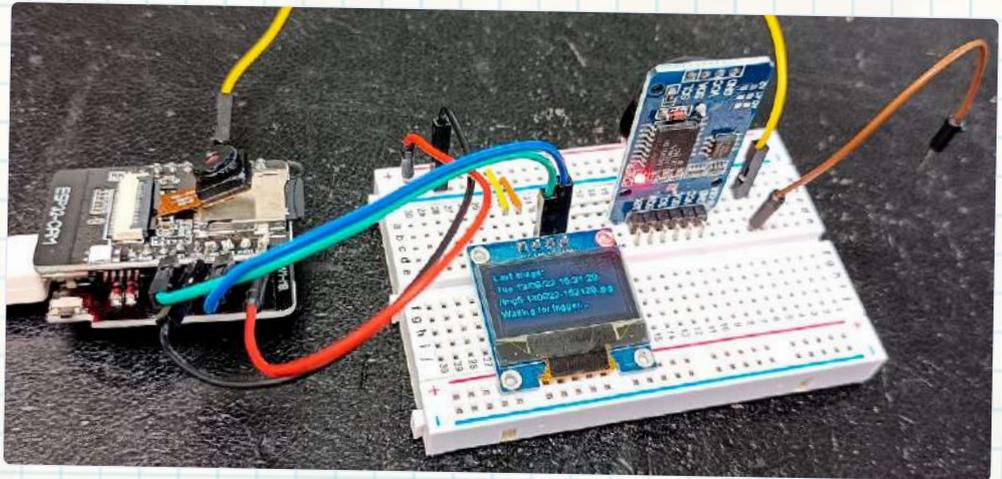


Bild 5. Der Prototyp von Elektor Labs, aufgebaut auf einem Breadboard. Die ESP32-Cam mit Tochterplatine ist links, das vertikale Modul rechts ist die RTC.

47 k $\Omega$  und einen Pull-down-Widerstand von 10+1-k $\Omega$ , der einen Transistor ansteuert. Port 13 besitzt ebenfalls einen Pull-Up-Widerstand von 47 k $\Omega$  und wird in diesem Projekt als Kamera-Trigger-Eingang verwendet.

IO12 ist einer der ESP32-Pins, die beim Einschalten mit Vorsicht behandelt werden müssen, da er den Wert von VDD\_SDIO und damit die Spannung der Flash-Speicherschnittstelle bestimmt. Er muss beim Einschalten auf dem ESP32-Cam-Board auf Low gezogen werden.

Nach einigem Experimentieren hat sich herausgestellt, dass der I<sup>2</sup>C-Bus sicher auf die Ports IO0 (SDA) und IO3 (SCL) oder umgekehrt (solange die Software richtig konfiguriert ist) gelegt werden kann. Bei der Verwendung dieser Ports stört der I<sup>2</sup>C-Bus weder den seriellen Programmierport noch den Boot-Modus, da sowohl IO0 als auch IO3 beim Einschalten auf High sein sollen.

Schließlich, und das hat nichts mit dem oben Gesagten zu tun, sollten Sie wissen, dass IO33 mit einer roten LED auf der ESP32-Modulseite des Boards verbunden ist, die Sie auf beliebige Weise verwenden können (solange Sie daran denken, dass der Port aktiv low ist).

Beim zweiten Projekt haben wir uns ein wenig ausgetobt und das Programm geändert (**Bild 6**). Es verwendet nun nicht mehr das emulierte EEPROM zum Speichern der Bildnummer, sondern eine Datei auf der SD-Karte. Löschen Sie einfach die Datei *counter.txt*, um den Zähler zurückzusetzen. Dieses unser Programm kann von [4] heruntergeladen werden.



Bild 6. Es funktioniert, Daumen hoch!

\* Eine weitere Version des ESP32-Cam-Moduls finden Sie unter [Passende Produkte](#).

# ATX-Netzteil für Raspberry Pi

Von Sébastien Guerreiro de Brito (Frankreich)

Haben Sie einen Raspberry Pi, der mit Strom versorgt werden muss, und ein altes ATX-Netzteil? Mit einer Platine, einigen Bauteilen und einem ATtiny können Sie beides miteinander verbinden.



Bild 1. ATX-Netzteil. Quelle: Shutterstock.

Das PC-Netzteil im ATX-Formfaktor (**Bild 1**) wurde 1995 von Intel eingeführt. Seit den Pentium-II-Prozessoren ist es der am weitesten verbreitete Formfaktor für Schaltnetzteile in PCs. Alle elektronischen, mechanischen, umweltrelevanten und anderen Eigenschaften sind von Intel spezifiziert worden, um den ATX-Standard zu definieren.

Der Zweck dieser Schaltung ist die Verwendung eines ATX-Netzteils zur Versorgung einer Raspberry-Pi-Platine und zahlreicher angeschlossener Peripheriegeräte. Der Vorteil dieses Netzteils ist die Zahl der verschiedenen Spannungen, die zur Verfügung steht - bei interessanten Ausgangsleistungen.

## ATX-Netzteilstecker

In diesem Artikel wollen wir nicht auf die interne Funktionsweise des ATX-Schaltnetzteils eingehen, sondern lediglich versuchen, es zum Laufen zu bringen. Betrachten

wir zunächst im Detail den Steckerverbinder in **Bild 2**, der an das Motherboard des PCs angeschlossen wird.

Der Steckerverbinder besteht aus

- einem 24-poligen, zweireihigen Mini-Fit Jr-Buchsengehäuse von Molex (39-01-2240) oder einem Äquivalent
- 24 Mini-Fit HCS-Buchsenkontakten von Molex (44476-1112)

Die Pinbelegung dieses Steckers ist in **Tabelle 1** zu sehen.

Die Signale, die für die Steuerung der Stromversorgung von Interesse sind, sind PWR\_OK und PS\_ON#. Die +5-VSB-Versorgung wird in unserem Projekt auch verwendet.

Das PWR\_OK-Signal ist ein „Power good“-Signal, mit dem die Stromversorgung dem System mitteilt, dass die Ausgangsspannungen +5 VDC, +3,3 VDC und +12 VDC

vorhanden sind und die Spannungen den Nennwerten entsprechen.

Das PS\_ON#-Signal ist das Signal, mit dem wir die Stromversorgung starten können. Es ist ein TTL-kompatibles Signal, aktiv-low, das es der Hauptplatine ermöglicht, die Stromversorgung fernzusteuern. Damit ist ein softwaregesteuertes Ein- und Ausschalten beispielsweise bei Wake-on-LAN oder Wake-on-USB möglich.

Wenn PS\_ON# auf Low gezogen wird, muss das Netzteil die vier Haupt-Gleichspannungsschienen aktivieren: +12 VDC, +3,3 VDC, +5 VDC und -12 VDC. Wenn PS\_ON# auf TTL-High gezogen wird oder offen bleibt, dürfen die DC-Ausgänge keinen Strom mehr liefern und müssen auf Null-Potential gegenüber Masse gehalten werden. Die Elektronik des ATX-Netzteils verfügt über eine so genannte interne Anti-Bounce-Schaltung, um Ein- und Ausschaltprellen zu verhindern,



Bild 2. Der ATX-Netzteilstecker.  
Quelle: Shutterstock.

Pin	Signal	Wire color
1	+3,3 VDC	Orange
2	+3,3 VDC	Orange
3	COM	Black
4	+5 VDC	Red
5	COM	Black
6	+5 VDC	Red
7	COM	Black
8	PWR_OK	Grey
9	+5 VSB	Pink
10	+12 V1DC	Yellow
11	+12 V1DC	Yellow
12	+3,3 VDC	Orange
13	+3,3 VDC	Orange
14	-12 VDC	Blue
15	COM	Black
16	PS_ON#	Green
17	COM	Black
18	COM	Black
19	COM	Black
20	Reserved	NC
21	+5 VDC	Red
22	+5 VDC	Red
23	+5 VDC	Red
24	COM	Black

Tabelle 1: Pinbelegung des ATX-Anschlusses

wenn das PS\_ON#-Signal durch einen mechanischen Schalter (de-)aktiviert wird. Die Eigenschaften des PS\_ON#-Signals sind in **Tabelle 2** aufgeführt und in **Bild 3** als Diagramm dargestellt.

Das PS\_ON#-Signal hat keinen Einfluss auf den Ausgang +5 VSB. Die +5-VSB-Versorgung (SB = Standby) ist immer vorhanden, wenn das Netzteil an das Stromnetz angeschlossen ist. Dieser Ausgang liefert Strom für Schaltungen, die in Betrieb bleiben müssen, auch wenn die Hauptspannungsschienen deaktiviert sind.

### Beschreibung der Elektronik

Damit das ATX-Netzteil vom Raspberry Pi gesteuert werden kann, wurde eine Platine entworfen [1], die die erforderliche Logik in Form eines kleinen Mikrocontrollers zum Starten und Herunterfahren des Raspberry Pi enthält. Dies bringt auch einige andere Vorteile mit sich. Da wir schon über ein ATX-Netzteil verfügen, können wir die 5-V-Schiene auch gleich nutzen, um eine Reihe von USB-Ports mit „Hochstromeigenschaften“ der Platine hinzuzufügen. Die mit dem Netzteil verbundene Platine ist in **Bild 4** zu sehen.

Parameter	Minimum	Maximum
$V_{IL}$	0 V	0,8 V
$I_{IL}$	-	-1,6 mA <sup>1</sup>
$V_{IH}$	2,0 V	-
$V_{IH, \text{offen}}$	-	-5,25 V
Restwelligkeit Rauschen		400 mV <sub>SS</sub>

<sup>1</sup> Hinweis: Negativer Strom bedeutet, dass Strom von der Stromversorgung zum Motherboard fließt.

Tabelle 2: PS\_ON#-Signaleigenschaften

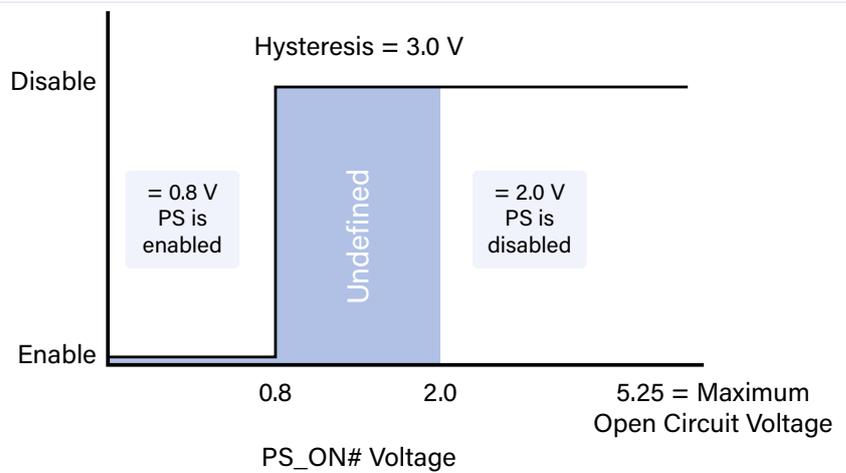
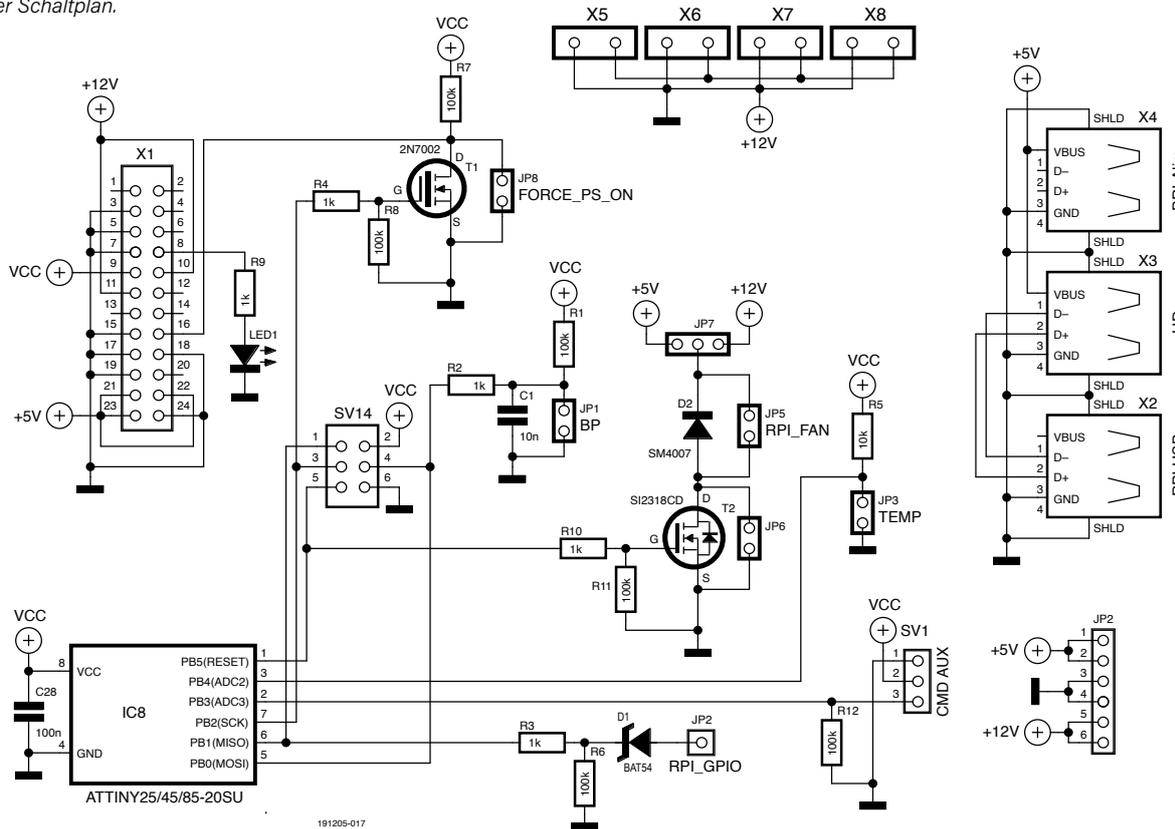


Bild 3. Eigenschaften des PS\_ON#-Signals. (Quelle: ATX/ATX12V Power Supply Design Guide Version 1.1, Abschnitt 3.3.2, Abbildung 3)



Bild 4. ATX-Netzteil, angeschlossen an die Platine.

Bild 5. Der Schaltplan.



Die recht einfache Schaltung zur Steuerung des ATX-Netzteils in **Bild 5** basiert auf einem ATtiny85-Mikrocontroller, der direkt von +5 VSB versorgt wird. Der Vorteil einer Mikrocontroller-Steuerung liegt darin, dass er trotz geringen Aufwandes die Möglichkeit bietet, weitere Funktionen hinzuzufügen.

### Einschalten der ATX-Netzteils

Der ATtiny wird über einen externen Taster an Jumper JP1 angewiesen, die Stromversorgung

zu starten. Sobald der Taster betätigt wird, steuert der Mikrocontroller den Transistor T1 an, der die Stromversorgung startet. Mit Jumper JP8 kann die Stromversorgung auch im automatischen Modus gestartet werden.

### Temperaturmanagement des Raspberry-Pi-Prozessors

Eine zusätzliche Lüfterkühlung des Raspberry-Pi-Boards wird durch einen simplen NTC-Thermistor (10 kΩ) am Pi-Controller

realisiert. Je nach gemessener Temperatur aktiviert die Zusatzschaltung einen Lüfter. Mit dem Jumper JP7 wählen wir eine Lüfterspannung von entweder 12 V oder 5 V.

### Kommunikation mit dem Raspberry Pi

Um das ATX-Netzteil wie bei einem richtigen PC vom Raspberry Pi aus abschalten zu können, haben wir uns einen kleinen Trick ausgedacht. Das Signal RPI\_GPIO wird mit GPIO27 auf der Pi-Platine verbunden. Der GPIO-Anschluss des Raspberry Pi ist in **Bild 6** zu sehen. Auf Ebene des Betriebssystems muss zunächst sichergestellt werden, dass GPIO27 zugänglich ist. Dazu erstellen wir mit dem nano-Editor die GPIO-Initialisierungsdatei, die beim Systemstart ausgeführt wird:

`nano S75gpioinit`

Dann geben wir den Code aus **Listing 1** ein. Wir machen die Datei ausführbar und legen sie im Initialisierungsordner ab.

```
sudo chmod +x S75gpioinit
sudo mv S75gpioinit /etc/init.d
```

Sie werden im Initialisierungsskript bemerken, dass GPIO27 mit einem Wert von 1 initialisiert ist. Wenn wir also den Raspberry Pi über das Betriebssystem ausschalten (Poweroff zum

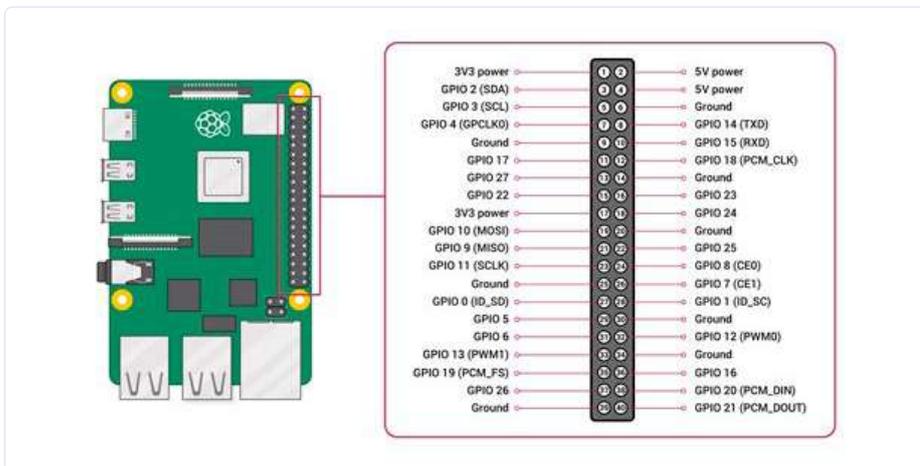


Bild 6. Pinbelegung des Raspberry Pi (Quelle: [2]).

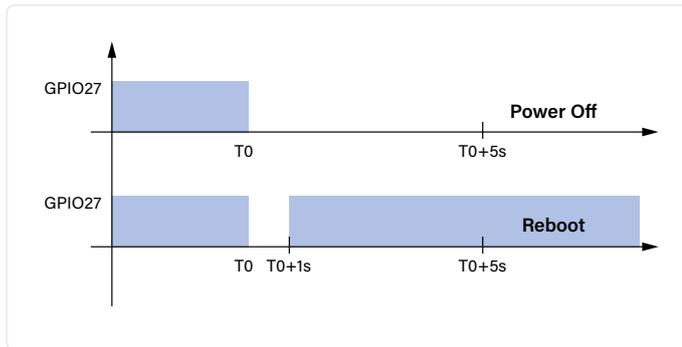


Bild 7. Verwaltung des GPIO27-Pins.

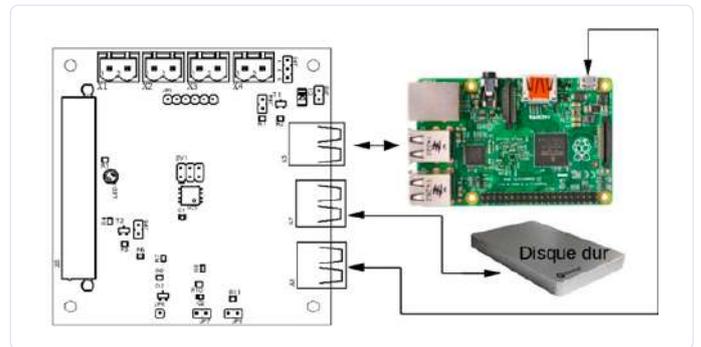


Bild 8. Anschließen des Raspberry Pi.

Beispiel), wird dieser Eintrag wieder auf 0 gesetzt und der Mikrocontroller weiß, dass das ATX-Netzteil auszuschalten ist. Um aber noch ein wenig mehr Luxus zu genießen, wollen wir einen Neustart des Raspberry-Pis über den Reset des Netzteils ermöglichen. Um dies zu erreichen, ändern wir den GPIO27, um dem Mikrocontroller zu

signalisieren, dass kein Ausschalten, sondern ein Reset erforderlich ist. Wie Sie in **Bild 7** sehen können, sieht die Verwaltung von GPIO27 im Einzelnen wie folgt aus: Wenn der Mikrocontroller sieht, dass der GPIO27-Pin auf 0 steht, wartet er fünf Sekunden lang. Wenn der Pin dann immer noch auf 0 steht, bedeutet das, dass

das RPi ausgeschaltet wurde, also schaltet er die ATX-Stromversorgung aus. Wenn nicht, empfängt er einen Reboot-Befehl und veranlasst das ATX-Netzteil, sich für zwei Sekunden abzuschalten und dann wieder neu zu starten. Damit dies alles möglich ist, müssen wir mit einigen Linux-Befehlen eingreifen. Auf der Seite des Betriebssystems benennen wir



### Listing 1: Shell-Script

```
#!/bin/sh
#####
# IO Init Script #
# Author : Sebastien Guerreiro #
# Versions : #
# ----- #
# Mars 2019 V1.0 Création #
#####
CHEM_GPIO=/sys/class/gpio
#-----
register()
{
#Registers the outputs
if [ $SENS="out" ]; then
#Outputs
echo "$NUM" > $CHEM_GPIO/export
echo "$SENS" > $CHEM_GPIO/gpio$NUM/direction
echo "$VALEUR" > $CHEM_GPIO/gpio$NUM/value
chmod g+w $CHEM_GPIO/gpio$NUM/value
else
#Inputs
echo "$NUM" > $CHEM_GPIO/export
echo "$SENS" > $CHEM_GPIO/gpio$NUM/direction
fi
ret=$?;
if [ $ret -eq 0 ]; then
#echo_success;
echo "Registering GPIO$NUM : OK"
else
# echo_failure;
echo "Registering GPIO$NUM : ERROR"
exit $ret;
fi
}
#-----
unregister()
{
echo "$NUM" > /sys/class/gpio/unexport
ret=$?;
if [ $ret -eq 0 ]; then
echo_success;
else
echo_failure;
exit $ret;
fi
}
#-----
start()
{
echo "Registering GPIO--Setting Power GPIO ON"
#Output for power GPIO
NUM=27; SENS=out ; VALEUR=1
register;
}
#-----
start
exit 0
```



### Listing 2: Das Shutdown-Script

```
#!/bin/sh
#####
# Reboot Poweroff Management #
# Author : Sebastien Guerreiro (www.sebelectronic.com)#
# Versions : #
# ----- #
# Fev 2020 V1.0 Création #
#####
case $1 in
-r)
echo 0 > /sys/class/gpio/gpio27/value
sleep 1
echo 1 > /sys/class/gpio/gpio27/value
sleep 1
/sbin/shutdownSeb $@
;;
*)
/sbin/shutdownSeb $@
;;
esac
```

die in `sbin` enthaltenen Anweisungen von `shutdown` und `reboot` in `shutdownSeb` beziehungsweise `rebootSeb` um

```
sudo mv /sbin/shutdown /sbin/shutdownSeb
sudo mv /sbin/reboot /sbin/rebootSeb
```

Anschließend erstellen die beiden Skripte `shutdown` und `reboot`, die wir im Ordner `/sbin` ablegen. Geben Sie im Terminal

```
nano shutdown
```

ein und geben Sie das **Listing 2** in die Datei ein. Speichern Sie diese Datei und beenden Sie nano. Geben Sie im Terminal ein:

```
sudo cp shutdown /sbin
```

Dadurch wird Ihr Skript nach `/sbin` kopiert, so dass Sie es später ausführen können. Als nächstes müssen wir das `reboot`-Skript ausführen. Geben Sie ebenfalls im Terminal

```
nano reboot
```

ein und kopieren Sie das Skript in **Listing 3** in diese Datei. Speichern Sie die Datei und beenden Sie nano. Geben Sie dann im Terminal ein:

```
sudo cp reboot /sbin
```

Dadurch wird Ihr `reboot`-Skript nach `/sbin` kopiert, so dass es später ausgeführt werden kann. Das war's!

### Anschließen der Peripherie

Wenn Sie ein ATX-Netzteil verwenden, können Sie das Board zum Beispiel an eine IDE-Festplatte oder in meinem Fall an ein IDE-CD-ROM-Laufwerk anschließen. Kurz gesagt, der Vorteil ist, dass Sie die Peripheriegeräte Ihres alten PCs für Ihre neuen Projekte verwenden können. Der Anschluss erfolgt wie in **Bild 8**.



### Listing 3: Das Reboot-Script

```
#!/bin/sh
#####
# Reboot Management #
# Author : Sebastien Guerreiro #
# Versions #
# ----- #
# Fev 2020 V1.0 Création #
#####
echo 0 > /sys/class/gpio/gpio27/value
sleep 1
echo 1 > /sys/class/gpio/gpio27/value
sleep 1
/sbin/rebootSeb
/sbin/shutdownSeb -r now
```



### PASSENDE PRODUKTE

- > **Raspberry Pi 4 2 GB (SKU 18965)**  
[www.elektor.de/18965](http://www.elektor.de/18965)
- > **Temperaturgesteuerte Lötstation Velleman VTSS220 (SKU 19865)**  
[www.elektor.de/19865](http://www.elektor.de/19865)

### WEBLINKS

- [1] Projekt auf Elektor Labs: [www.elektormagazine.de/labs/atx-powersupply-for-rpi](http://www.elektormagazine.de/labs/atx-powersupply-for-rpi)
- [2] Bildquelle: [www.raspberrypi.com/documentation/computers/images/GPIO-Pinout-Diagram-2.png](http://www.raspberrypi.com/documentation/computers/images/GPIO-Pinout-Diagram-2.png)

Wenn Sie mehr über das Projekt wissen möchten, können Sie die Elektor-Labs-Seite [1] besuchen. 

(191205-01)RG

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Wenden Sie sich bitte an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de) oder hinterlassen Sie einen Kommentar auf der Projektseite bei Elektor Labs.



## STÜCKLISTE

### Widerstände:

(alle 0805, 0,1 W)

R1, R3, R6, R7, R8 = 1 k, 1%

R2, R4, R5, R9, R10 = 100 k, 1%

R11 = 10 k, 1%

### Kondensatoren:

C1 = 100 n, 0805

C2 = 10 n, 0805

### Halbleiter:

D1 = SM4007

D2 = BAT54

IC1 = ATTINY85

T1 = SI2318CDS

T2 = 2N7002

LED1 = LED, grün, 3 mm

### Außerdem:

SV1, JP7 = 1x3-polige Stiftleiste, Raster 2,54 mm

JP2 = 1x1-polige Stiftleiste, Raster 2,54 mm

JP1, JP3, JP5, JP6, JP8 = 1x2-polige Stiftleiste,

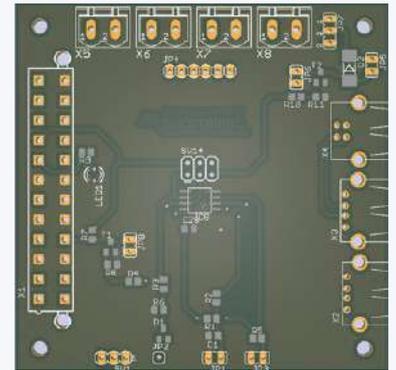
Raster 2,54 mm

SV14 = 1 St. 2x3-polige Stiftleiste, Raster 2,54 mm

X1 = Molex Mini-Fit Jr gerader Steckverbinder, 5566, 24-polig, 2 Reihen

X2, X3 = USB-Typ-A-Steckverbinder

X4 = USB-Typ-B-Steckverbinder



X5, X6, X7, X8 = 1x2-polige Platinen-Anschlussklemme MSTBVA 2,5/2-G-5,08 Platine 191205

Advertisement

# Treten Sie jetzt der Elektor Community bei!

- ✓ Komplettes Webarchiv ab 1970
- ✓ 8x Elektor Doppelheft (Print)
- ✓ 8x Digital (PDF)
- ✓ 10% Rabatt im Online-Shop und exklusive Angebote
- ✓ Zugriff auf über 5.000 Gerber Dateien aus Elektor Labs



Jetzt  Mitglied werden!



[www.elektormagazine.de/Abonnement](http://www.elektormagazine.de/Abonnement)

 **elektor**  
design > share > earn

# 32Ω

## Kopfhörerverstärker

Einfache, aber hochqualitative 3-Chip-Lösung

Von Thierry Clinquart (Belgien)

Früher hatten die meisten guten Kopfhörer eine Impedanz von 600 Ω. Heute dominieren in den mittleren und höheren Preiskategorien 32-Ω-Versionen. Deshalb brauchen moderne Kopfhörerverstärker etwas mehr Leistung.

Während ein normaler Operationsverstärker prinzipiell ausreicht, um 600-Ω-Kopfhörer sehr laut zu betreiben, fließen bei vergleichbarer Lautstärke in den Schwingspulen moderner 32-Ω-Kopfhörer deutlich höhere Ströme, die normale Operationsverstärker übersteuern würden. Die klassische Lösung mit den alten, aber guten Audio-Opamps NE5532 und vergleichbaren ICs ist damit hinfällig. Damit moderne Kopfhörer nicht verzerrt klingen, muss also die klassische Opamp-Schaltung ausgetrickst werden. Glücklicherweise ist auch dies sehr einfach, nicht nur im Prinzip, sondern auch in der Praxis, denn alles, was Sie hinzufügen müssen, ist ein BUF634A pro Kanal.

### Der BUF634A

Im Grunde ist der BUF634A eine kleine und schnelle Push-Pull- oder Gegentakt-Endstufe in einem IC-Gehäuse. Man muss ihn nur hinter einen Operationsverstärker schalten und in die Gegenkopplung einbeziehen - fertig!

**Bild 1** zeigt die komplette Stereoschaltung. Ein Dual-Audio-Opamp OPA2134 [1] bildet die beiden Eingangsstufen und ist für die Spannungsverstärkung zuständig. Zwei BUF634As übernehmen die Stromverstärkung. Seine interne Grundschaltung ist in **Bild 2** zu sehen.

Die Besonderheit dieser integrierten Gegentakt-Ausgangsstufe ist die sehr hohe Bandbreite, die über den Strom durch Pin 1 zwischen 35 MHz und 210 MHz eingestellt werden kann. Lässt man den Widerstand Rx in Bild 1 weg (Pin 1 = offen), beträgt die Bandbreite 35 MHz und der Ruhestrom nur etwa 1,5 mA. Die maximale Bandbreite ergibt sich, wenn für Rx ein Wert von 0 Ω gewählt wird. In diesem Fall steigt

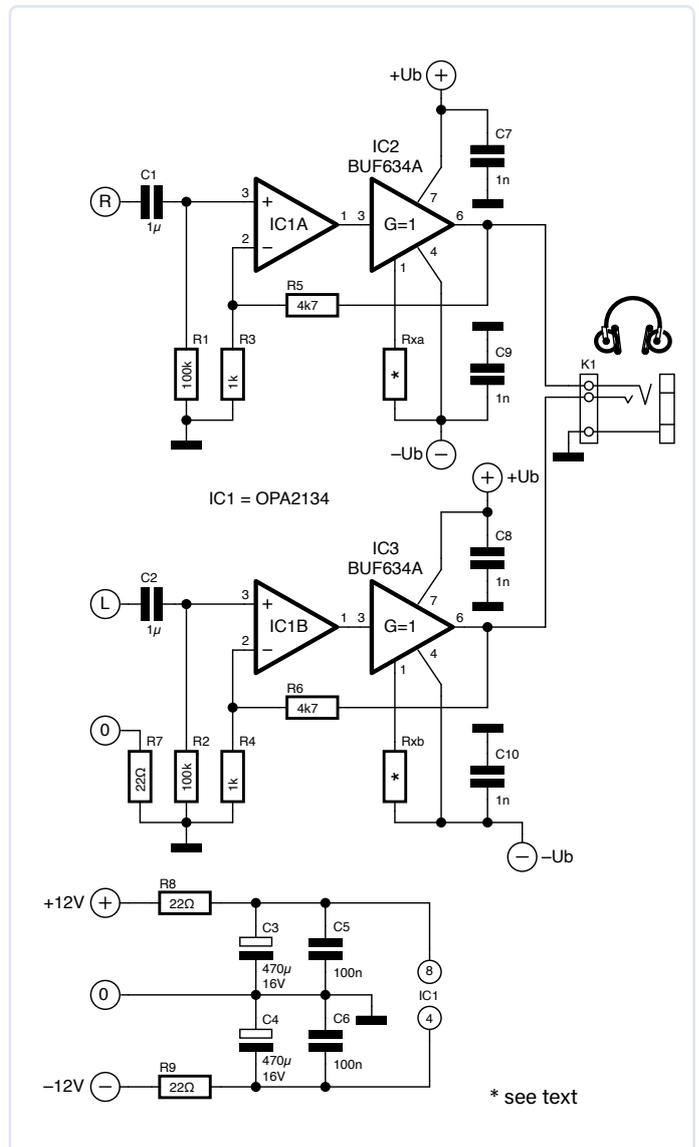


Bild 1. Die Schaltung des kompletten Kopfhörerverstärkers basiert auf integrierten Halbleitern.

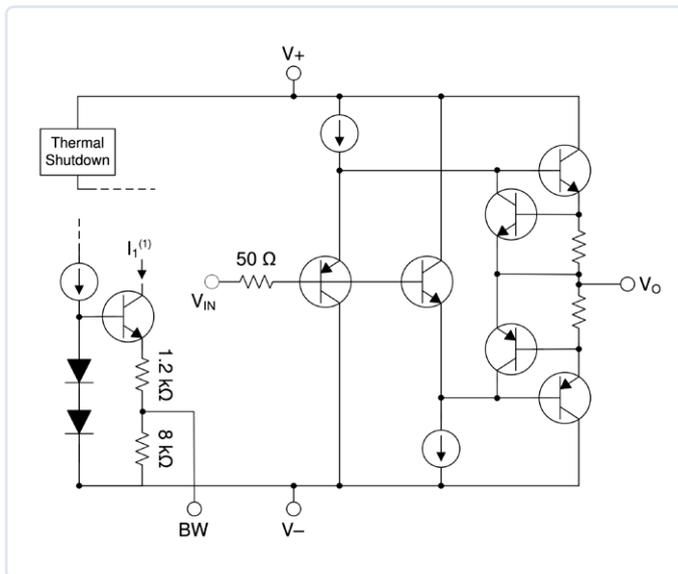


Bild 2. Die Innenschaltung der integrierten Gegentakt-Ausgangsstufe BUF634A. Quelle: [2].

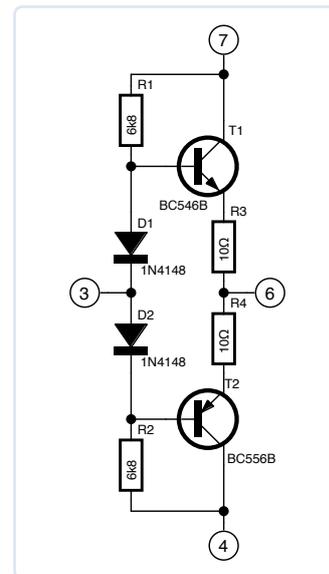


Bild 3. Diese diskrete Schaltung kann als Ersatz für den BUF634A IC verwendet werden.

der Ruhestrom auf tolerierbare 8,5 mA an. Die Ausgangsstufe kann Ströme von bis zu 250 mA liefern – sogar genug für Kopfhörer mit Impedanzen bis hinunter zu 8 Ω. Detaillierte Informationen entnehmen Sie bitte dem Datenblatt [2].

Doch welche Bandbreite und damit welcher Ruhestrom ist die richtige? Für die Anwendung als kleiner Kopfhörerverstärker sollte eine geringe Bandbreite normalerweise in Ordnung sein, so dass man auf Rx verzichten kann. Für höhere kapazitive Lasten ist jedoch der höhere Ruhestrom mit  $R_x = 0 \Omega$  die bessere Wahl. Letzteres gilt auch für Operationsverstärker, die bei Einfachverstärkung nicht stabil sind. Ruhestrome zwischen den Extremen sind mit anderen Werten von  $R_x$  leicht einstellbar.

### Feinabstimmung und Modifikation

Mit SMD-Bauteilen lässt sich ein hervorragender Kopfhörerverstärker mit minimaler Verzerrung, sehr geringem Rauschen und hoher Bandbreite auf kleinstem Raum bauen. Aber auch diejenigen, die bedrahtete Bauteile bevorzugen, werden mit dieser Schaltung glücklich. IC1 ist nämlich auch in einer 8-poligen DIP-Version erhältlich, und für IC2 und IC3 kann man problemlos als Ersatz den Typ BUF634 verwenden, der im 8-poligen DIP-Gehäuse erhältlich ist. Dieser Typ ist jedoch etwas langsamer, weist einen etwas höheren Ruhestrom auf und wird deshalb für neue Designs nicht mehr empfohlen.

Eine weitere Möglichkeit besteht darin, den integrierten BUF634A durch eine diskrete Ersatzschaltung aus Kleinsignaltransistoren zu ersetzen, wie in **Bild 3** geschehen. Der Ruhestrom hängt dann von den jeweiligen kombinierten Eigenschaften der Halbleiter ab und kann über die Werte von R1 und R2 eingestellt werden. Für die thermische Stabilität ist außerdem darauf zu achten, dass D1 und D2 in direktem Kontakt mit T1 und T2 stehen. Falls erforderlich, können sie auch verklebt werden.

Da der Ruhestrom auch von der Versorgungsspannung abhängt, sollte der Verstärker mit einer stabilisierten  $\pm 12\text{-V}$ -Versorgung betrieben

werden, zum Beispiel mit den Spannungsregler-ICs 7812 und 7912. Für 32-Ω-Kopfhörer ist eine Stromversorgung von  $\pm 100\text{ mA}$  ausreichend, für 8-Ω-Kopfhörer wird etwa das Doppelte benötigt. R7 reduziert Brummeinstreuungen durch mögliche Masseschleifen.

Sie können natürlich auch mit anderen Typen für IC1 oder T1 und T2 experimentieren und gute Ergebnisse erzielen. Bei Kopfhörern mit Impedanzen  $\geq 32 \Omega$  gibt es mit dem BUF634A keine Wärmeprobleme.



### STÜCKLISTE

#### Widerstände:

(alle Metallfilm, 1%)

R1,R2 = 100 k

R3,R4 = 1 k

R5,R6 = 4k7

R7..R9 = 22 Ω

Rxa,Rxb = siehe Text

#### Kondensatoren:

C1,C2 = 1 μ, 25 V, Folie

C3,C4 = 470 μ, 16 V, Elko

C5,C6 = 100 n, 25 V

C7..C10 = 1 n, 25 V, Keramik

#### Halbleiter:

IC1 = OPA2134

IC2,IC3 = BUF634A (siehe Text)

#### Außerdem:

K1 = Kopfhörer-Klinkenbuchse, Stereo, 6,3 mm



Bei 8  $\Omega$  und längerem Betrieb bei hoher Lautstärke können SOIC-ICs aber recht warm werden. Achten Sie beim Entwurf einer Platine deshalb darauf, dass die Wärme über die Pads an eine ausreichend große Kupferfläche abgeleitet wird, oder wählen Sie das 8-polige DRB-Gehäuse, das auf der Unterseite ein „Thermal Die Pad“ hat. Der Typ BUF634 (ohne das „A“) ist sogar in einem leicht zu kühlenden TO-220- oder TO-263-Gehäuse erhältlich.

Für die Eingangskondensatoren C1 und C2 verwenden Sie am besten Folientypen. Am Ausgang sind keine Kondensatoren erforderlich, da die typische Offset-Spannung bei wenigen Millivolt liegt, was keine nennenswerte Belastung der Schwingspulen darstellt. Wenn Sie jedoch absolut sicher sein wollen, können Sie ja je einen Kondensator zwischen Ausgang und Kopfhörer schalten. Für eine Impedanz von 32  $\Omega$  wäre ein bipolarer Kondensator von 470  $\mu\text{F}$  (25 V) ausreichend, 8- $\Omega$ -Kopfhörer benötigen dagegen mindestens 1000  $\mu\text{F}$ . Der Ausgang ist kurzschlussfest - allerdings nicht, wenn Sie den BUF634-Ersatz aus Bild 3 verwenden.

## Fazit

Moderne Analogelektronik ermöglicht kleine Kopfhörerverstärker mit ausgesprochen audiophilen Eigenschaften. Klirrfaktoren im Bereich von <0,01 % bei mittleren Lautstärken sind ohne weiteres erreichbar und das Rauschen ist dank der guten Eigenschaften des IC1 praktisch unhörbar. Der Verstärker kann bis zu 300 mW an 32- $\Omega$ -Kopfhörern abgeben (und bei guter Kühlung immer noch bis zu 250 mW an 8- $\Omega$ -Lasten), was viel mehr ist, als man seinen audiophilen Ohren zumuten sollte. ◀

(200441-02)RG

## Über den Autor

Als gelernter Elektroniker hat Thierry Clinquart seine Leidenschaft auf die Audiotechnik gelenkt. Alle seine Projekte drehen sich um analoges Audio: Vorverstärkung, dynamische Verarbeitung, Korrekturen, Signalverteilung und so weiter. Er fertigt seine eigenen Platinen mit Sprint-Layout von Abacom und nutzt sPlan für die Schaltpläne.

## Haben Sie Fragen oder Kommentare?

Bei technischen Fragen wenden Sie sich bitte per E-Mail an die Elektor-Redaktion unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



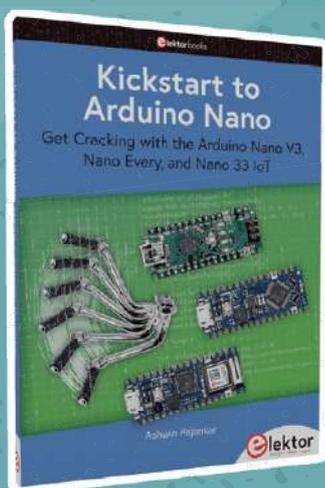
## Passende Produkte

- > **2-Kanal-Oszilloskop OWON SDS1102 (100 MHz) (SKU 18782)**  
[www.elektor.de/18782](http://www.elektor.de/18782)
- > **PeakTech 3350 True RMS Digitalmultimeter (6000 Counts) (SKU 19986)**  
[www.elektor.de/19986](http://www.elektor.de/19986)
- > **Joy-IT JDS6600 Signal Generator und Frequenzzähler (SKU 18714)**  
[www.elektor.de/18714](http://www.elektor.de/18714)

## WEBLINKS

[1] Datenblatt OPA2134: <https://ti.com/product/OPA2134>

[2] Datenblatt BUF634A: <https://ti.com/product/BUF634A>



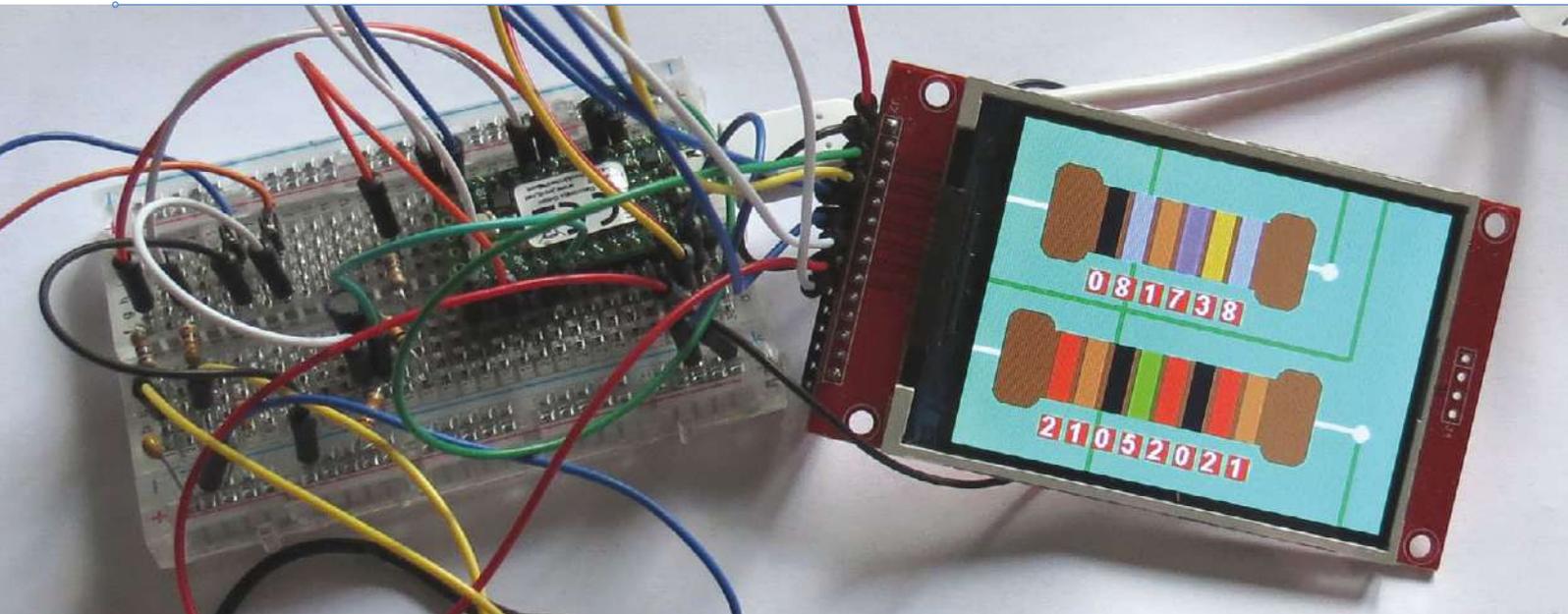
## Kickstart to Arduino Nano

Das Buch dient als erster Schritt für Einsteiger und Mikrocontroller-Enthusiasten, die einen Einstieg in die Arduino-Programmierung suchen. Das Buch folgt einem schrittweisen Ansatz, um Konzepte und die Funktionsweise von Dingen zu vermitteln. Jedem Konzept folgen ein Schaltplan und Code-Beispiele, die auf den Leser gebracht werden. Danach folgen detaillierte Erklärungen der Syntax und der Logik. Wenn Sie die Konzepte genau verfolgen, werden Sie sich mit dem Aufbau von Schaltungen, der Arduino-Programmierung, der Funktionsweise von Code-Beispielen und den dargestellten Schaltplänen vertraut machen.

[www.elektor.de/20241](http://www.elektor.de/20241)



elektor



Das Mikrocontrollerboard Teensy 4.0 mit angeschlossenen Display auf einer Steckplatine bildet SDR-Funkuhr. Die Uhrzeit wird hier im Widerstands-Farbcode angezeigt.

# SDR-Funkuhren

Fünf Zeitzeichen, sechs Anzeigen

Von Prof. Dr. Martin Oßmann

Heutige Mikrocontroller sind so leistungsfähig, dass man mit ihnen Funkuhren nach dem Prinzip des Software Defined Radio aufbauen kann. Wie das prinzipiell funktioniert, hat die MSF-Funkuhr [1] mit einem Raspberry Pi Pico gezeigt. Und auch in diesem Artikel geht es um Zeitsignale, aber als Mikrocontroller wird das schnelle Teensy-4.0-Board eingesetzt.

**Tabelle 1. Für diese Sender sind Empfänger realisiert.**

60,0 kHz	MSF	britischer Zeitzeichensender NPL
77,5 kHz	DCF77	Zeitzeichen Deutschland
129,1 kHz	EFR	Funk-Rundsteuerung
162 kHz	TDF	französisches Zeitzeichen
198 kHz	BBC	AM-Rundfunk mit Phasenmodulation

Wie entwickeln in diesem Projekt insbesondere einen Empfänger für die Signale des EFR-Funkdienstes, der zur Rundsteuerung von den Energieversorgern eingesetzt wird und in regelmäßigen Abständen auch die Zeitinformation sendet. Eine Funkuhr damit zu bauen ist etwas Neues. Das gleiche SDR-basierte Konzept und die gleiche Hardware kann man auch zum Empfang anderer Zeitzeichensignale verwenden, was wir ausgiebig nutzen wollen (**Tabelle 1**). Ein weiteres wichtiges

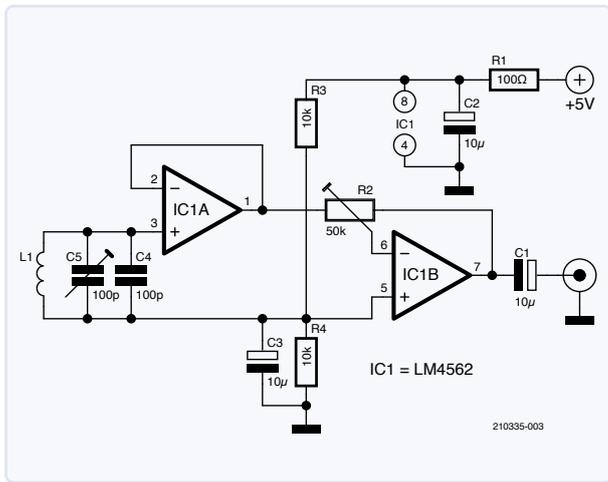


Bild 1. Schaltplan der aktiven Antenne für 129,1 kHz. Die 250 Windungen des 0,2 mm<sup>2</sup> starken Kupferlackdrahtes sind auf einen ETD29-Spulenkörper gewickelt. Die Spule wird auf einen 150x8 mm messenden Ferritstab aus dem Material 3B1 geschoben. Durch Anpassung von C<sub>x</sub> ist die Schaltung benutzbar für Frequenzen von 60 kHz bis 200 kHz.

Entwicklungsziel ist die Anzeige der Zeit auf verschiedenen innovativen Displays.

### Ein Konzept für alle Frequenzen

Zum Empfang der Funksignale ist erst einmal eine aktive Antenne nach dem Schaltplan in **Bild 1** erforderlich. Das Signal des Ferritstabes (150x8 mm, Material 3B1, Spulenkörper ETD29, Wicklung 250 Windungen CuL 0,2 mm<sup>2</sup>) wird mit einem Schwingkreis gefiltert und dann von einem Opamp gepuffert. Als zweite Stufe folgt ein weiterer Opamp, dessen Verstärkung mit dem Trimpoti R2 an die örtlichen Gegebenheiten angepasst werden kann. Am besten arbeitet unser Empfänger, wenn am Ausgang der zweistufigen Verstärkungsschaltung ein Signal von rund 50 mV<sub>SS</sub> anliegt. Die Schaltung ist für alle hier genutzten Zeitzeichen-Frequenzen geeignet; will man die Antenne für andere Frequenzen benutzen, muss man den Kondensator C4 anpassen. Das Schaltbild des Gesamtsystems sehen Sie in **Bild 2**. Hauptkomponente ist ein Teensy-4.0-Board. Mit den Widerständen R1 und R2 stellen wir den Offset des A/D-Wandlers ein und koppeln mit C1 das Signal der aktiven Antenne auf den A/D-Wandler des Teensy-Boards. Wir haben unseren Prototypen auf einem Steckboard aufgebaut, wie in Bild 1 zu sehen ist. Die beiden Tiefpässe R4/C2 und R3/C3 filtern die PWM-Signale von Ausgang 6 und Ausgang 7. Die PWM-Signale verwenden wir zum Debugging, um beispielsweise Signal I und Signal Q auf einem Oszilloskop darzustellen. Das Display ist über den SPI-Anschluss und einige zusätzliche Pins angeschlossen. Die Stromversorgung erfolgt über den USB-Anschluss, über den das Board auch programmiert wird und die

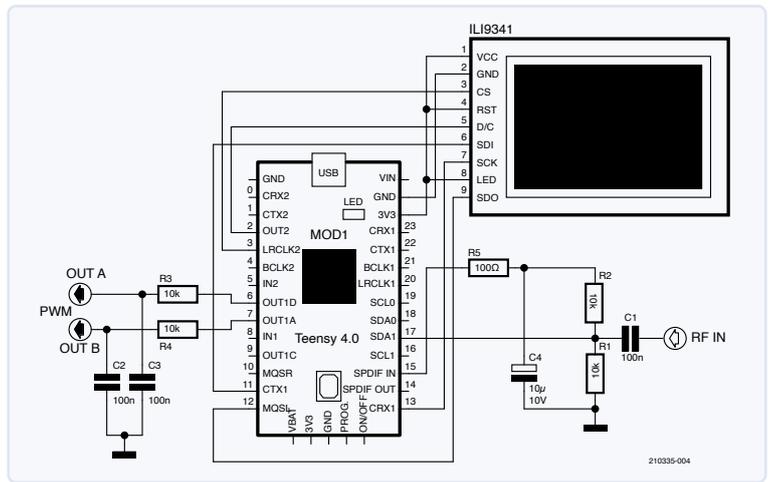


Bild 2. Schaltplan unseres Software Defined Radio für Funkuhren.

Debugging-Texte ausgegeben werden. So kann man im seriellen Monitor der Arduino-Entwicklungsumgebung den Empfang der Daten mitverfolgen. Das Teensy-Board lässt sich in der normalen Arduino-Entwicklungsumgebung programmieren.

### Zeitzeichen-Empfang

Das „universelle“ Konzept bezieht sich nicht nur auf die Hardware, sondern auch auf die Software des Empfängers. Sie ist in ihrer Struktur mehr oder weniger für alle hier aufgeführten Zeitzeichensender ähnlich. Da aber die Demodulations- und Dekodier-routinen doch sehr verschieden voneinander sind, erschien es mir sinnvoller, für jeden Zeitzeichensender eine separate Firmware zu schreiben, als alle Routinen in eine einzige Firmware zu packen und beispielsweise per Switch die gewünschte zu aktivieren. Das komplette und dokumentierte Software-Paket kann von der Projektseite unter [2] heruntergeladen werden. Die prinzipielle Arbeitsweise des Empfängers soll aber zuerst am Empfang des EFR-Longwellensenders auf 129,1 kHz ausführlicher beschrieben werden.

### EFR-Empfang

Die EFR GmbH betreibt drei Langwellensender (129,1 kHz, 139 kHz, 135,6 kHz) zur Rundsteuerung bei Energieanlagen [3]. Die Sender haben jeweils eine Reichweite von wenigstens 500 km und decken einen großen Teil Mitteleuropas ab. Als Modulationsart wird Frequency-Shift-Keying (FSK) mit ±170 Hz Hub und 200 Bit/s Datenrate verwendet. Neben den verschlüsselten proprietären Nachrichten wird auch die Zeitinformation gesendet. Das Protokoll ist in IEC 60870-5 festgelegt.

Das Konzept unseres EFR-Empfängers ist in **Bild 3** dargestellt. Das Signal der aktiven Antenne gelangt an den internen, mit 333.333 kHz getakteten ADC des Teensy4.0-Boards. Damit ist das Abtasttheorem erfüllt. Wie bei SDR-Empfängern üblich, erfolgt dann eine IQ-Multiplikation. Das Oszillatorsignal wird mit einem in der Software realisierten DDS-Generator erzeugt. Nach der Multiplikation des Eingangssignals mit dem I- beziehungsweise dem Q-Oszillatorsignal erfolgt eine Tiefpassfilterung. Die Tiefpassfilter sind so dimensioniert, dass das Empfangssignal

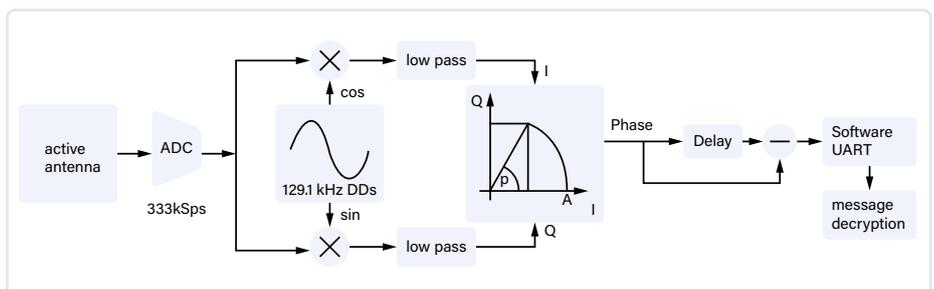


Bild 3. Das Konzept des EFR-Empfängers.

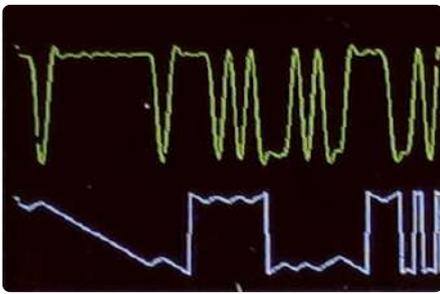


Bild 4. Debug-Anzeige auf dem Display. Obere Spur: Demoduliertes RTTY-Signal (Momentanfrequenz), untere Spur: Momentanphase.

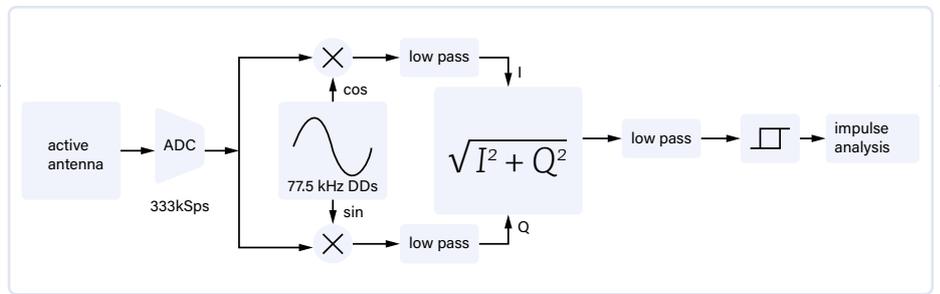


Bild 5. Konzept des DCF-Empfängers.

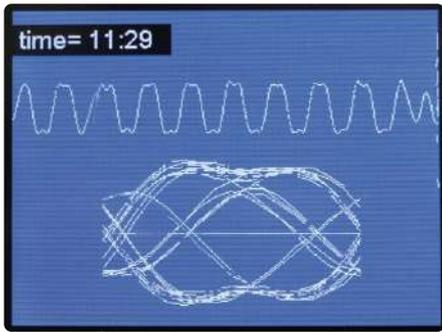


Bild 6. Debug-Informationen beim BBC-Empfang. Oben: Zurückgewonnenes Datensignal, unten: Augendiagramm.

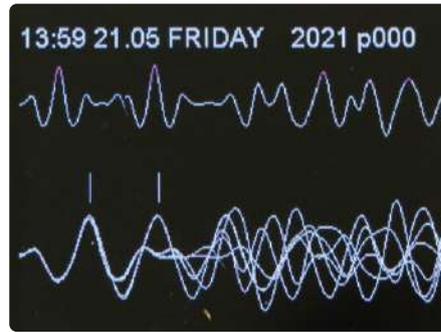


Bild 7. Debug-Signale beim TDF-Empfang.

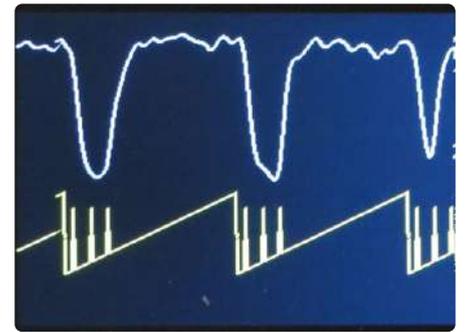


Bild 8. MSF-Debuginfo. Oben: Empfangspegel mit Sekundenmarken, unten: Timer-Variablen mit Abtastpunkten.

noch nicht beschnitten wird, aber auch keine weiteren Signalanteile passieren können. Um das RTTY-Signal zu demodulieren, müssen wir die aktuelle Empfangsfrequenz bestimmen. Bei +170 Hz Offset handelt es sich um eine Eins, bei -170 Hz um eine Null. Dazu bestimmen wir aus den I/Q Signalen die aktuelle Phase. Das machen wir mit der  $\text{atan2}()$ -Funktion. Aus der zeitlichen Phasenänderung können wir die Frequenz bestimmen. Dazu bilden wir die Differenz aus Original und verzögertem Signal und gelangen so zum demodulierten Signal, das wir auf ein Software-UART geben, um die gesendeten Bytes zurückzugewinnen. Damit erlangt man die gesendeten Nachrichten zurück, aus denen man die Zeit entnehmen kann. **Bild 4** zeigt die Debug-Signale auf dem Display, oben das demodulierte RTTY-Signal (Momentanfrequenz) und unten die Momentanphase.

### DCF-Empfang

Das Konzept des DCF-Empfängers (**Bild 5**) ähnelt stark dem EFR-Konzept. Diesmal wird allerdings aus den I/Q-Signalen die momentane Sendeamplitude bestimmt. Dazu berechnen wir:  $\sqrt{I^2 + Q^2}$ . Dieses Signal wird tiefpassgefiltert auf einen Schmitt-Trigger geleitet. Damit stehen die Sekundenimpulse zur Auswertung zur Verfügung und wir können Datum und Uhrzeit aus den Bits bestimmen.

### BBC-Empfang

Die BBC sendet auf 198 kHz ein Rundfunkprogramm aus. Der Träger wird phasenmoduliert und zur Datenübertragung ähnlich dem RDS-System bei FM benutzt [4]. Die Phasenmodulation wird genauso demoduliert wie beim EFR-System. Aus dem demodulierten Signal (**Bild 6** oben) muss der Bittakt von 25 Bit/s zurückgewonnen werden. Dazu bildet der Empfänger sozusagen das Augendiagramm und variiert den Takt so, dass das Auge sich maximal öffnet. In der Mitte des Auges werden dann die Daten abgetastet. Aus dem so erhaltenen Bitstrom muss man nun die Blockgrenzen zurückgewinnen. Die Daten werden nämlich jeweils in Blöcken mit je 50 Bit gesendet, davon sind jeweils 13 Bit Prüfbits. Der Empfänger sucht im Bitstrom nach fehlerfreien Blöcken und erkennt an ihrer Lage die Blockgrenzen. Es gibt verschiedene Blocktypen, die verschiedene Daten enthalten. Blocktyp 0 ist zur Übertragung der Zeitinformation vorgesehen und wird von unserem Empfänger ausgewertet.

### TDF-Empfang

Bis 2016 sendete die TDF (ehemals Télédiffusion de France) ein Rundfunkprogramm auf 162 kHz, jetzt wird nur noch der Träger mit der Phasenmodulation ausgesendet. In den Daten ist unter anderem auch die Zeitinformation enthalten. Ein Null-Bit wird als Einzel-

impuls, ein Eins-Bit als Doppelimpuls gesendet, die jeweils am Sekundenbeginn anfangen. In **Bild 7** sieht man die empfangene Phaseninformation (obere Spur) und unten als Art Augendiagramm übereinandergelegte Spuren, die links zwischen den senkrechten Linien die Impulse zeigen. In der 59. Sekunde nach dem Minutenbeginn ist das Signal nicht phasenmoduliert. Diese Pause benutzt unser Empfänger zur Synchronisation. Die einzelnen Bitinformationen sind ganz ähnlich aufgebaut wie beim DCF77.

### MSF auf 60 kHz

Auf 60 kHz betreibt das NPL (National Physical Laboratory) in Großbritannien einen Langwellensender, der mit 15 kW die Zeitinformation aussendet. Es wird die einfache Trägereinstufung verwendet [5]. Die Minute beginnt mit einer Pause von 500 ms. Die nachfolgenden Sekunden beginnen jeweils mit einer Trägerabsenkung von 100 ms und dann weiteren zwei Absenkungen von gegebenenfalls 100 ms, in Abhängigkeit der zu übertragenen Bits. Unser MSF-Empfänger ist ganz ähnlich dem DCF-Empfänger aufgebaut, nur die Pulsdekodierung erfolgt anders. Verglichen mit den anderen Signalen ist das MSF-Signal relativ schwach, aber richtig tief im Westen, in Aachen, noch ganz gut zu empfangen (**Bild 8**).

## Anzeige-Optionen

Als sich der Verfasser entschloss, eine EFR-basierte Funkuhr zu bauen, musste auch entschieden werden, welches Display zum Einsatz kommen sollte. Die Entscheidung fiel dann zugunsten eines 2,8"-LC-Displays mit 320x240 Pixel, womit man die verschiedensten Displayoptionen realisieren kann. Das Display basiert auf dem Controller ILI9341 und kann zum Beispiel von Roboter-Bausatz.de bezogen werden. Alternativ lässt sich der kleine Bruder des Displays mit einer Diagonalen von 2,2", aber gleicher Auflösung aus dem Elektor-Store (siehe Textbox Passende Produkte) einsetzen. Das Display ist über die SPI-Schnittstelle mit dem Teensy-Board verbunden.

### Widerstands-Display

Unsere EFR-Clock bietet eine etwas ungewöhnliche neuartige Displayoption, das sogenannte Widerstands-Display an (Bild 9). Die Zahlen von Uhrzeit und Datum werden dabei farbkodiert wie bei elektrischen Widerständen angezeigt. Im Display finden sich zwei Widerstände, einer mit sechs Ringen für die sechs Ziffern der Uhrzeit (Stunden:Minuten:

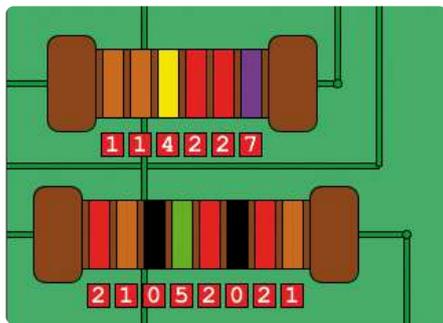


Bild 9. Displayoption Widerstandscode.

Sekunden, also beispielsweise 12:23:45). Der zweite Widerstand weist acht Ringe für das Datum auf (Tag.Monat.Jahr, zum Beispiel 23.08.2012). Die Uhrzeit wird sekundlich aktualisiert, und die lokale Uhr wird immer synchronisiert, wenn vom EFR-Dienst eine Uhrzeit empfangen wird. Für Benutzer, die den Widerstandscode noch nicht kennen, wird die Uhrzeit und das Datum auch mit klassischen Ziffern angezeigt. Puristische Leser werden hier darauf verwiesen, dass Widerstände höchstens sechs Ringe besitzen, aber immerhin kann so der Unkundige anschaulich den DIN41429-Code der Farbbrünge erlernen.

### Domino-Display

Als weiteres lustiges Display wurde eine Domino-Anzeige implementiert (Bild 10). Auf Dominosteinen kann man ebenfalls die Zahlen von 0..9 darstellen. Auf einem Dominostein gelingt die Darstellung von zwei Ziffern, so dass man für die Darstellung der Uhrzeit drei und für das Datum vier Dominosteine benötigt. Ob das Widerstandsdisplay, das Dominodisplay oder eine andere Option zum Einsatz kommt, entscheidet man im

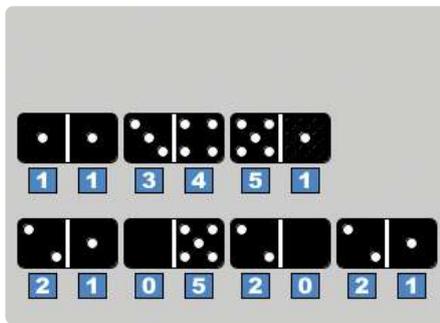


Bild 10. Domino-Display.

Quellcode durch Optionsbits (Listing 1). Die zu wählende Option wird mit einer „1“ angewählt, die anderen Optionen mit einer „0“ ausgeschaltet.

### Word-Clock und Analog-Style

Zusätzlich zu den bisher beschriebenen Displays kann man auch ein Word-Clock-Display wählen, in dem die Uhrzeit sozusagen als Klartext im Display dargestellt wird (Bild 11). Bei dieser Option wird nur die Uhrzeit



#### Listing 1. Durch diese Konstanten legt man das Display fest.

```
#define scopeShow 0
#define scopeClkShow 1
#define resShow 0
#define dominoShow 0
#define wordShow 0
#define sevenSegShow 0
#define AP6571Show 0
```



Bild 11. Auch eine Word-Clock ist implementiert.

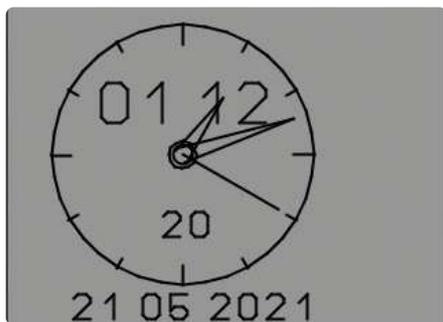


Bild 12. Klassische Analog-Uhr.



Bild 13. Siebensegment-Display ganz nostalgisch.



Bild 14. Die U(h)rzeit wie aus dem 9-Nadel-Drucker.

angezeigt, nicht das Datum. Eine Aktualisierung erfolgt alle fünf Minuten. Auch die klassische Darstellungsform einer Analoguhr ist implementiert (**Bild 12**). Die Uhrzeit wird auf einem runden Zifferblatt-Kreis dargestellt, das Datum wird als Text darunter angezeigt.

### Sieben-Segment-Display

Ein einfaches nostalgisches Siebensegment-Display haben wir natürlich auch im Angebot. Wer also an die alten Zeiten erinnert werden will: Das ist die richtige Wahl (**Bild 13**).

### Zeichengenerator-Display

Als letzte Option bieten wir ein einfaches Text-Display basierend auf dem legendären Zeichengenerator-ROM 6571AP, das einst vielfach in 9-Nadel-Druckern eingesetzt wurde (**Bild 14**).

Insgesamt stellt dieses Projekt eine große Anzahl von Kombinationsmöglichkeiten und Optionen zur Verfügung. Man kann zwischen den fünf Funkuhrvarianten von EFR bis DCF und zwischen sechs Anzeigeoptionen wählen. Da müsste eigentlich für jeden etwas dabei sein. Wer will, kann die vorliegende Software,

zum Beispiel die Displayoptionen, auch als Ausgangspunkt für eigene Entwicklungen nehmen. **◀**

210335-02

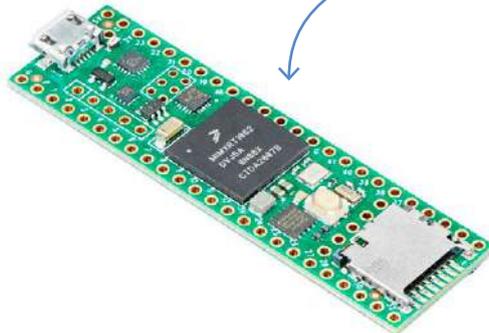
### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Projekt? Dann nehmen Sie Kontakt mit dem Autor [ossmann@fh-aachen.de](mailto:ossmann@fh-aachen.de) oder mit der Elektor-Redaktion [redaktion@elektor.de](mailto:redaktion@elektor.de) auf!



### Passende Produkte

- > **2.2" SPI TFT Display-Modul ILI9341 (240x320) (SKU 18419)**  
[www.elektor.de/18419](http://www.elektor.de/18419)
- > **Teensy 4.1 Development Board (SKU 19311)**  
[www.elektor.de/19311](http://www.elektor.de/19311)



### WEBLINKS

- [1] Martin Ossmann, „Raspberry Pi Pico als SDR für das MSF-Zeitsignal“, ElektorMag 7-8/2022: <https://www.elektormagazine.de/magazine/elektor-265/60575>
- [2] Firmware auf der Projektseite: <https://www.elektor.de/210335-02>
- [3] EFR-Signal: [https://www.mee.hu/files/images/5/B\\_Sbick\\_EFR-CEE\\_Lakihegy.pdf](https://www.mee.hu/files/images/5/B_Sbick_EFR-CEE_Lakihegy.pdf)
- [4] Spezifikation L.F. Radio Data:  
<https://downloads.bbc.co.uk/rd/pubs/reports/1984-19.pdf>
- [5] MSF 60 kHz Time and Date Code: [https://www.npl.co.uk/products-services/time-frequency/msf-radio-time-signal/msf\\_time\\_date\\_code](https://www.npl.co.uk/products-services/time-frequency/msf-radio-time-signal/msf_time_date_code)



**LPN liefert Leiterplatten** aus Deutschland, vom Weltmarkt, aus NATO-Partnerländern oder mit anderen Restriktionen.

**LPN** ist nach ISO 9001:2015 Zertifiziert und das Personal beim FraunhoferInstitut geschult.

**LPN** liefert jedes Basismaterial und jede in Deutschland oder am Weltmarkt verfügbare Technik.

- Multilayer bis 56 Lagen.
- Starrflex, Flex, Semiflex.
- Aluminium, auch Bergquist, Kupferkern, Messingkern, Stahlkern.
- Teflon, auch Rogers.
- Montagehilfen Kaptonband, Abziehlack und Weiteres.

### LPN Qualitätsprüfungen

- 100% Kontrolle
- Kupferstärkenmessung mit Magnetfeld Messgeräten.
- Nachmessen gedruckter Induktivitäten.
- Schliffbildauswertung.
- Lot-Benetzungs-Test.
- Delaminations-Test.
- alle Fertigungsstätten halten ISO 14001 ein.

### LPN Dienstleistungen

- Datenaufbereitung incl. Nutzenaufbau,
- Machbarkeitsprüfung,
- EMPB.
- geklebte Vorlagen digitalisieren.
- Filme digitalisieren.
- Leiterplatten clonen.
- Leiterplatten nachlayouts.
- Terminaufträge.
- Abruflager für Jahreslose.

Profitieren Sie von den LPN Qualitätsstandards und den weltweiten Kontakten.

### LPN Leiterplatten Nord GmbH

Hermann-Bössow-Straße 13-15  
23843 Bad Oldesloe  
[leiterplatten-nord.de](http://leiterplatten-nord.de)

### Anfragen/Bestellungen:

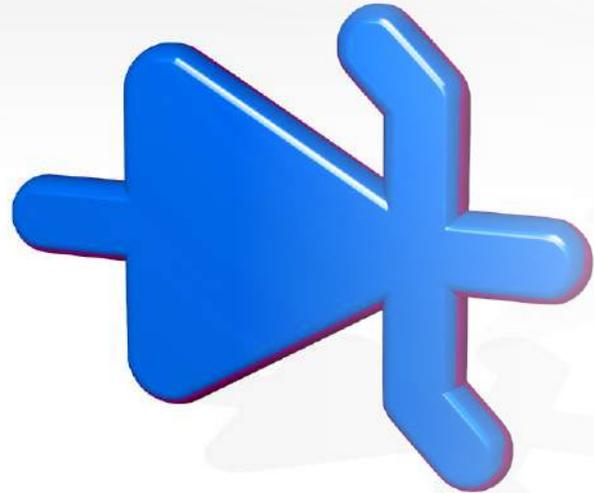
[lpn@lp-nord.de](mailto:lpn@lp-nord.de)  
Telefon 04531 1708 0

# Aller Anfang ...

muss nicht schwer sein:  
Spezielle Dioden

Von Eric Bogers (Elektor)

In diesem Teil schließen wir das Kapitel mit einigen eher exotischen Dioden ab, die auch heutzutage noch genutzt werden, wie Thyristoren, Triacs, LEDs und Optokoppler.



Die erste Generation von Dimmerschaltungen wurde mit Thyristoren gebaut - Triacs gab es damals noch nicht (zumindest nicht für die benötigten Leistungen). Heutzutage werden in Dimmern fast ausschließlich Triacs verwendet.

## Der Thyristor

Ein Thyristor (**Bild 1**) ist im Grunde nichts anderes als eine Diode mit einem Steueranschluss - wenn der Thyristor leitet, dann nur in Durchlassrichtung, genau wie eine normale Diode. Für Wechselstromanwendungen benötigen wir daher zwei Thyristoren (oder einen Gleichrichter).

Um einen Thyristor in den leitenden Zustand zu versetzen, verfügt er über einen dritten Anschluss, das so genannte *Gate* (die beiden anderen Anschlüsse heißen *Anode* und *Kathode*, genau wie bei einer normalen Diode), an dem er einen „Zündimpuls“ benötigt. Ist ein Thyristor einmal gezündet, leitet er weiter, solange ein Strom durch ihn fließt – die Spannung am Gate muss nicht aufrechterhalten werden. Um den Thyristor auszuschalten und in den vorherigen Zustand zurückzukehren, muss der Stromfluss durch ihn unterbrochen werden. In einem Gleichstromkreis leitet ein Thyristor daher „für immer“, in einem Wechselstromkreis nur bis zum nächsten Nulldurchgang.

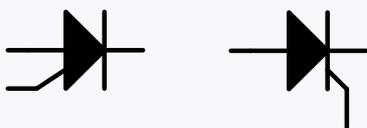


Bild 1. Der Thyristor.

Es gibt zwei Möglichkeiten, einen Dimmer mit einem Thyristor zu konstruieren: Die Wechselspannung kann gleichgerichtet werden, oder es können zwei Thyristoren antiparallel geschaltet werden. Da die so konstruierten Dimmerschaltungen keine praktische Bedeutung mehr haben, wird hier nicht weiter darauf eingegangen. Thyristoren werden aber häufig als Überspannungsschutz eingesetzt, wie in **Bild 2** dargestellt. Wenn nämlich ein Spannungsregler oder ein Leistungstransistor im Netzteil durchbrennt, dann liegt ohne Schutz eine (viel) zu hohe Spannung an den angeschlossenen Bauteilen und Geräten an, die diese zerstören kann. Das ist eine Situation, die man unbedingt vermeiden sollte.

Sobald jedoch die Spannung in der Schaltung von Bild 2 zu hoch wird, leitet die Z-Diode und zündet den Thyristor, der daraufhin ständig leitet. Dadurch entsteht ein Kurzschluss, der zum Auslösen der Sicherung führt. Dies ist aber das geringere Übel im Vergleich zu einer völlig zerstörten elektronischen Schaltung.

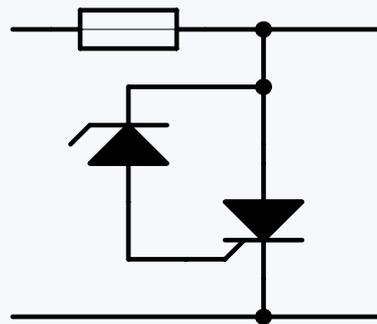


Bild 2. Prinzip des Überspannungsschutzes.

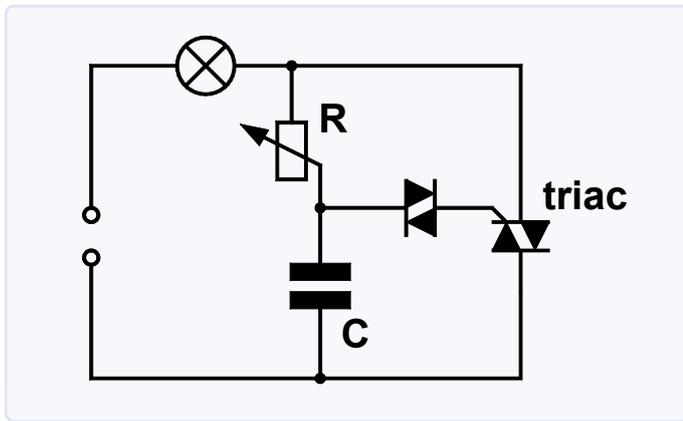


Bild 3. Konventionelle (sprich: veraltete) Dimmerschaltung.

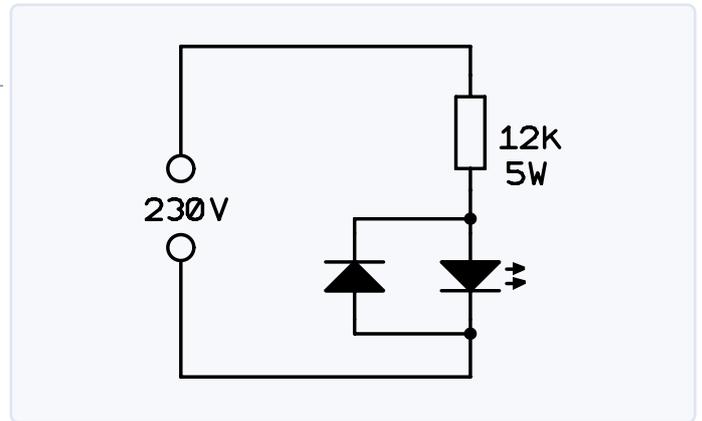


Bild 5. LED mit Flyback-Diode.

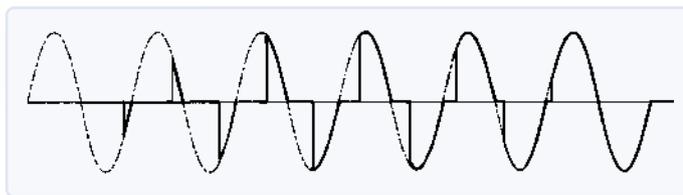


Bild 4. Phasenanschnitt.

Je nach Typ des Thyristors benötigt dieser zur Zündung eine Spannung von 0,8...2,5 V. Die Z-Spannung sollte daher sorgfältig gewählt werden, damit dieser Zündspannungspegel im Fall einer Überspannung auf jeden Fall erreicht wird.

### Der Triac

Genau wie ein Diac, der im Grunde eine Kombination aus zwei Z-Dioden ist, ist ein Triac eine Kombination aus zwei Thyristoren - er existiert ausschließlich zu dem Zweck, Wechselstromanwendungen mit nur einem Bauteil zu realisieren. Schauen wir uns den Triac in einer praktischen Schaltung an - der herkömmlichen Dimmerschaltung, wie in **Bild 3** dargestellt. Diese Art von Dimmer arbeitet nach dem Prinzip des Phasenabschnitts: Je nachdem, ob ein kleinerer oder größerer Teil der Phase (Periode) durchlaufen wird, leuchtet die angeschlossene Lampe weniger oder mehr hell (**Bild 4** zeigt weitere Einzelheiten).

Der Triac schaltet sich ein, wenn die Spannung am Diac über etwa 33 V steigt und den Zündimpuls für den Triac liefert. Wenn das Potentiometer R in Bild 3 auf den niedrigsten Wert eingestellt ist, wird die Spannung einerseits durch das RC-Netzwerk geteilt und andererseits um 90° in der Phase verschoben - die Spannung über dem Diac erreicht nie einen Wert von 33 V. Wird das Potentiometer anschließend allmählich verstellt, steigt die Spannung über dem Diac (und auch die Phasenverschiebung wird immer geringer), mit dem Ergebnis, dass der Triac zu einem früheren Zeitpunkt zündet. Die Schaltung in Bild 3 ist einer der einfachsten (Netzspannungs-) Dimmer, den man sich vorstellen kann: Es fehlt jegliche Form der Entstörung, und auch für eine Schalthysterese wurden keine Maßnahmen ergriffen. Heutzutage gibt es viel bessere Dimmerschaltungen, aber sie sind auch viel komplizierter.

Natürlich könnte man eine Glühbirne dimmen, indem man ein einstellbares (Leistungs-)Potentiometer in Reihe schaltet, aber ein großer Teil der Wechselstromleistung würde in diesem Potentiometer als Verlustwärme verloren gehen. Der größte Vorteil des Phasenabschnitts ist also, dass dieser Leistungsverlust nicht auftritt.

### Licht emittierende Dioden und Optokoppler

Leuchtdioden sind Dioden, die Licht aussenden, sobald ein Strom durch sie fließt. Die Durchlassspannung beträgt etwa 2 V, während der maximale Strom normalerweise etwa 20 mA beträgt (bei so genannten Low-current-LEDs liegt dieser maximale Strom im Bereich von nur 2 mA).

LEDs hatten schon vor vielen Jahren normale Glühlampen in Schalttafeln ersetzt. Sie zeichnen sich durch eine sehr lange Lebensdauer, einer relativ geringen Stromaufnahme und eine hohe Stoß- und Vibrationsfestigkeit aus.

Heutzutage gibt es LEDs in allen möglichen Farben und auch Laser-LEDs (denen wir den CD- und DVD-Player verdanken), während Neopixel-LEDs mit eingebautem Mikrocontroller die Ausgabe aller Farben und auch verschiedener Lichtfarben ermöglichen. Auch in der Beleuchtungstechnik haben LEDs mit hoher Leistung als Ersatz für Glühbirnen Einzug gehalten.

Obwohl das alles sehr interessant ist, würde eine ausführliche Diskussion dieser Themen den Rahmen dieses Grundkurses bei weitem sprengen. Wir werden uns daher auf eine kurze allgemeine Diskussion beschränken, um dieses kleine, aber sehr wichtige Bauteil zu würdigen.

### Mit Wechselspannung betriebene LEDs

In vielen Fällen beträgt die zulässige LED-Sperrspannung nur wenige Volt. Wenn sie mit Wechselspannung betrieben werden, muss eine Flyback-Diode angeschlossen werden, um die Sperrspannung an der LED auf einen sicheren Wert zu begrenzen (siehe **Bild 5**). Übrigens: Das Schaltplansymbol einer LED ist eine normale Diode, allerdings mit einem Doppelpfeil, der ihre Lichtemission symbolisiert.

### Optokoppler

Ein Optokoppler besteht aus einer Leuchtdiode und einem lichtempfindlichen Bauteil (meist ein Fototransistor) in einem Gehäuse (**Bild 6**). Sobald eine Spannung an die Diode angelegt wird, leuchtet die Diode auf, wodurch der Fototransistor zum Leiten angeregt wird. Der große Vorteil dieses scheinbar komplizierten Ansatzes ist die galvanische Trennung zwischen LED und Fototransistor:

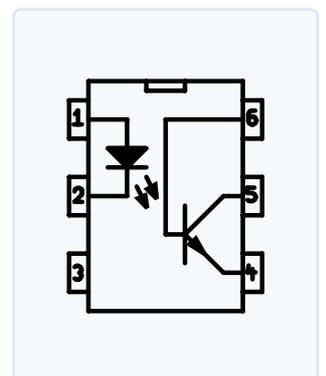


Bild 6. Optokoppler.

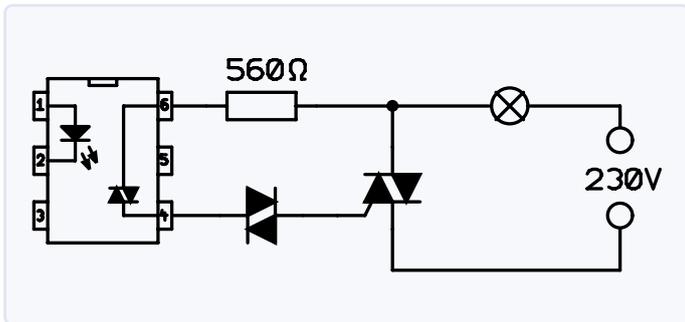


Bild 7. Optokoppler mit Phototriac.

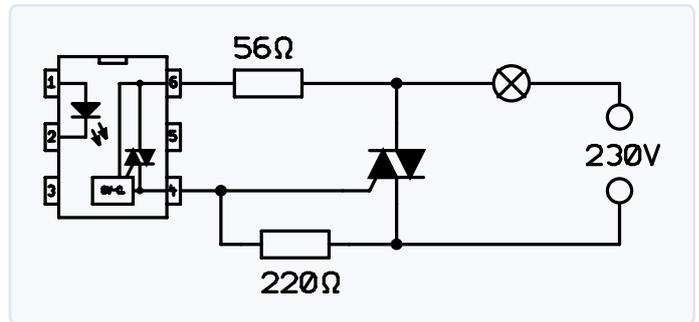


Bild 8. Optokoppler mit Nulldurchgangsdetektor.

Ein Spannungsunterschied von (oft) mehreren hundert Volt zwischen diesen Bauteilen ist problemlos möglich. Optokoppler werden häufig eingesetzt, um den Leistungsteil von Dimmern oder anderen Schaltungen von den Steuerschaltungen zu trennen. So kann zum Beispiel der Hochleistungsteil aus dem Netz gespeist werden, während (aus Sicherheitsgründen) die Steuerschaltung davon unbeeinflusst bleibt.

### Optokoppler mit Phototriac

Ein Transistor leitet nur in eine Richtung (wie wir in der nächsten Folge sehen werden), und das kann für manche Anwendungen ungünstig sein. Wenn wir damit einen Triac ansteuern wollten, bräuchten wir eine eigene Spannung, und der Triac würde mit einem Zündimpuls entgegengesetzter Polarität jede zweite Hälfte einer Wechselspannungsperiode ausgelöst, was zu einem asymmetrischen Verhalten führt. Um es kurz zu machen: Ein normaler Optokoppler ist für solche Anwendungen keineswegs ideal. Daher verwenden wir in solchen Fällen lieber einen Optokoppler mit einem Phototriac statt einem Phototransistor (**Bild 7**).

Wenn die LED mit Strom versorgt wird und leuchtet, beginnt der Phototriac zu leiten und schaltet den externen Triac. Bei leichten Lasten (zum Beispiel einem kleinen Lüfter) kann der externe Triac bei Bedarf entfallen und die Last direkt vom Phototriac gespeist werden. Die Schaltung in Bild 7 sollte auch ohne Widerstand und Diac funktionieren, doch es können zwei Probleme auftreten:

- Ohne den Diac neigt die Schaltung dazu, unkontrolliert auszulösen: Der Triac beginnt zu leiten, ohne dass ein Steuerimpuls von der LED kommt.
- Viele Leistungstriacs lassen den Phototriac durchbrennen, wenn der Strom nicht durch einen Widerstand begrenzt wird.

### Optokoppler mit Nulldurchgangsdetektion

Wenn der Strom an beliebiger Stelle der Stromphase eingeschaltet wird, erzeugt er einen hochfrequenten Störimpuls, der beispielsweise bei Audiosystemen hörbare Störungen verursachen kann. Diese HF-Störung ist die Ursache für das berühmte „Dimmer-Summen“. Leider haben Dimmer, die nach dem Prinzip des Phasenschnitts arbeiten, einen solchen Nachteil: Wir können nur versuchen, das Problem durch den Einsatz geeigneter Entstörfilter in den Griff zu bekommen.

Aber in allen Situationen, in denen eine Last nur ein- oder ausgeschaltet werden muss, sollte der Schaltzeitpunkt mit dem Nulldurchgang der Netzwechselspannung zusammenfallen: Dann treten Störungen erst gar nicht auf. Man kann dafür eigene ausgeklügelte Schaltungen entwerfen und bauen oder ... einen Optokoppler mit integrierter Nulldurchgangserkennung verwenden.

Natürlich ist es nicht möglich, damit einen Triac genau im Nulldurchgang zu schalten, weil dann keine Spannung zur Verfügung steht, die den benötigten Zündstrom liefert. Es ist jedoch möglich, so nahe wie möglich am Nulldurchgang zu schalten, was zu einer vernachlässigbaren Störung führt.

Damit schließen wir unsere Entdeckungsreise durch die wunderbare Welt der Dioden ab. In der nächsten Folge werden wir den Transistor vorstellen. ◀

(220446-02)RG

Die Artikelreihe „Aller Anfang ...“ gründet auf dem Buch „Basiskurs Elektronik“ von Michael Ebner, erschienen im Elektor-Verlag.

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



### Passende Produkte

- **B. Kainka, Elektronik-Grundlagen und Einsteiger-Projekte (Elektor 2019)**  
Buch, kartoniert, SKU 19035: [www.elektor.de/19035](http://www.elektor.de/19035)
- **B. Kainka, Elektronik-Grundlagen und Einsteiger-Projekte (Elektor 2019)**  
E-Buch, PDF, SKU 19036: [www.elektor.com/19036](http://www.elektor.com/19036)

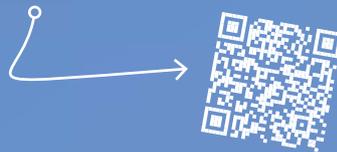
# Elektor TV Shows



## Elektor **Engineering Insights** (englische Show)

Elektor Industry Insights ist die erste Anlaufstelle für vielbeschäftigte Ingenieure und Hersteller, die sich über die Welt der Elektronik informieren möchten. In jeder Folge diskutiert Stuart Cording (Redakteur, Elektor) mit Experten aus der Elektronikbranche über echte technische Herausforderungen und Lösungen.

[www.elektormagazine.com/elektor-engineering-insights](http://www.elektormagazine.com/elektor-engineering-insights)



## Elektor **LabTalk** (englische Show)

Haben Sie eine Leidenschaft für DIY-Elektronik, eingebettete Systeme oder technische Aufgaben? Diskutieren Sie mit! Unsere Ingenieure und Redakteure Jens Nickel und Mathias Claußen geben in jeder Folge technische Tipps, besprechen zukünftige Elektronikprojekte, geben Einblicke in die Elektor Ausgaben und beantworten Fragen aus der Community.

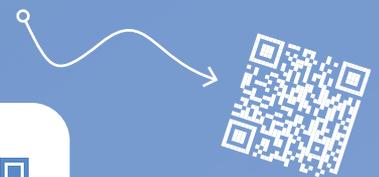
[www.elektormagazine.com/elektor-lab-talk](http://www.elektormagazine.com/elektor-lab-talk)



## **elektor academy** (englische Kurse)

Möchten Sie Ihre Elektronikkenntnisse verbessern? In der Elektor Academy finden Sie Kurse, um Ihre technischen Fähigkeiten zu verbessern. Unser Experte Stuart Cording führt Sie dabei durch die Kurse der Elektor Academy.

[www.elektormagazine.com/elektor-academy](http://www.elektormagazine.com/elektor-academy)



Bleiben Sie informiert und abonnieren Sie unseren YouTube-Kanal Elektor TV:

[www.youtube.com/c/ElektorIM](http://www.youtube.com/c/ElektorIM)



# Von Entwicklern für Entwickler

Tipps & Tricks, Best Practices und andere nützliche Infos



Photo: Shutterstock/ Jackrit Singhania

## Aus dem Leben gegriffen

Über die Qualität der Dinge

Von Ilse Joostens (Belgien)

Oft hört man ältere Menschen sagen: „Früher war alles besser“. Sie beziehen sich auf die „gute alte Zeit“ und schimpfen manchmal auch auf die jungen Leute von heute. Ich bin seit über 50 Jahren auf der Welt und weiß, dass in der Vergangenheit nicht alles besser war. Der Gedanke, dass früher alles besser war, begleitet uns schon ewig und kommt hauptsächlich daher, dass wir uns an positive Erfahrungen besser erinnern als an negative.

Trotzdem glaube ich nicht, dass mein Gehirn mir einen Streich spielt, wenn ich das Gefühl habe, dass die Qualität von Konsumgütern früher besser war. Natürlich waren manche Dinge, die damals verkauft wurden, Schrott, aber die heutige Situation ist ziemlich

schlecht. Der technologische Fortschritt führt nicht immer zu Verbesserungen und mehr Möglichkeiten. Leider schränkt er auch die Freiheit der Verbraucher ein, und die heimtückische Verkürzung der Lebensdauer von Produkten macht die Sache noch schlimmer.

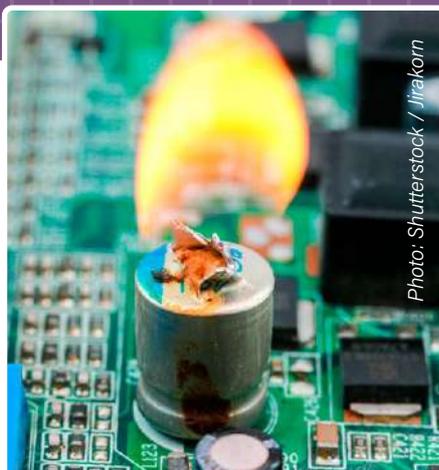
### Königliche Lampen und nicht austauschbare Batterien

Mit der Erfindung der hellen blauen LEDs im Jahr 1993 wurde eine Revolution in der Beleuchtungstechnik eingeleitet, auch wenn es eine Weile dauerte, bis LED-Lampen in den Geschäften erhältlich waren. Neben einem sehr niedrigen Stromverbrauch wurde uns eine erstaunliche Nutzungsdauer von mehreren zehntausend Stunden versprochen. In der Praxis fühlt es sich manchmal eher wie Minuten als Stunden an - ich habe aufgehört zu zählen, wie viele defekte LED-Lampen ich ersetzen musste. Unternehmen wollen einfach nur Produkte verkaufen, und Produkte mit einer langen Lebensdauer sind nicht gut für den Umsatz. In der Elektronikindustrie ist es allzu üblich geworden, Produkte am Rande der maximalen Belastbarkeit der Komponenten - oder sogar darüber hinaus - zu entwickeln. Das ist schwieriger, als Sie vielleicht denken, denn Produkte sollten möglichst bald nach Ablauf der Garantiezeit ausfallen, aber nicht früher, damit der Hersteller nicht für Garantieansprüche aufkommen muss. Leider fallen die Verbraucher immer wieder auf die gleichen falschen Versprechungen herein und kaufen neue Modelle in der Hoffnung, dass diese besser sind. So verhält es sich auch mit

LED-Lampen. Viele von ihnen sind so konzipiert, dass sie den maximal zulässigen Strom durch die LEDs jagen, oder sogar noch etwas mehr. Das Verhältnis zwischen Lichtleistung und Stromstärke ist nicht linear, so dass der Wirkungsgrad sinkt und mehr Strom in Wärme umgewandelt wird. Bei diesem Ansatz haben es sowohl die LEDs als auch die anderen Komponenten schwer, was zu einem vorzeitigen Ausfall der Lampe führt. Dann kann man eine neue Lampe kaufen: mehr Einkommen für die Lieferkette.

Das muss nicht so sein. Um die Stadt Dubai energieeffizienter zu machen, schloss Scheich Mohammad Bin Rashid Al Maktoum einen Vertrag mit Philips über die Produktion von Millionen von LED-Lampen, den so genannten Dubai-Lampen [1][2][3]. Der Betrieb der Lampen mit der Hälfte ihrer maximalen Nennleistung verlängert nicht nur ihre Lebensdauer erheblich, sondern sorgt auch dafür, dass diese Lampen bei gleicher Lichtleistung nur halb so viel Strom verbrauchen wie herkömmliche LEDs. Natürlich konnte man diese Lampen lange Zeit nicht im Laden kaufen. Erst vor kurzem hat Philips mit der *Master Ultra Efficiency Serie* vergleichbare LED-Lampen auf den Markt gebracht.

Vor Jahren wurde ich gebeten, einen Satellitenempfänger zu reparieren, der an einer Form von früherer Demenz litt und wahllos Zeichen anzeigte. Es stellte sich heraus, dass alle Elektrolytkondensatoren auf der 12-V-Stromschiene der Hauptplatine auf 10 V ausgelegt waren. Ich habe keine Ahnung, wie lange ein durchschnittlicher 10-V-Elko dem Betrieb bei 12 V standhält, aber das muss Absicht gewesen sein, denn die Garantie war gerade abgelaufen. Nachdem ich die Kondensatoren durch 16-V-Kondensatoren ersetzt hatte, funktionierte der Empfänger wieder einwandfrei. Das ist kein Einzelfall - in einem defekten



Akkuladegerät eines Marken-Rasenmähers fand ich einen 10-V-Elko im Stromversorgungs-Filterkreis, der Spitzenspannungen von bis zu 14 V verkraften musste.

Jeder weiß, dass Lithium-Ionen-Akkus nicht ewig halten, also sollten sie austauschbar sein. Das ist bei meinem kabellosen Staubsauger nicht möglich. Kurz nach Ablauf der Garantiezeit (schon wieder!) begann die Betriebszeit stark zu sinken. Nach viel Tüftelei, Fluchen und einem zerfetzten Finger habe ich es geschafft, die 18650er-Zellen zu ersetzen - aber der durchschnittliche Staubsaugerbenutzer ist kein Experte im Punktschweißen und Löten. Unser alter, stromfressender Zylinderstaubsauger funktioniert dagegen auch nach 24 Jahren noch gut. Ähnlich verhält es sich mit den neueren Smartphones, bei denen das Auswechseln des Akkus zunehmend einer Operation am offenen Herzen gleicht.

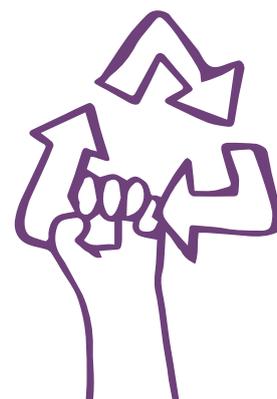
### Der Software-Gremlin und das Neue Europäische Bauhaus

All das ist nur „old school“, aber die Zahl der Möglichkeiten, die Verbraucher zu schikanieren, ist dank der zunehmenden Digitalisierung und des Einsatzes von Software nahezu unendlich geworden. Das reicht von Druckerpatronen mit eingebauten Chips bis zu endlosen Software-Updates, die Ihre Geräte (wie Computer oder Smartphones) langsamer oder sogar unbrauchbar machen, weil sie angeblich

veraltet sind. Oder noch schlimmer: An einem bestimmten Tag teilt Ihnen Ihr Drucker freudig mit, dass er das Ende seiner Lebensdauer erreicht hat und stellt sofort seinen Betrieb ein [4]. Ein bekannter Hersteller von Etikettendruckern [5] hat es sogar geschafft, die Etiketten mit RFID-Tags auszustatten, so dass man deren teures Papier kaufen muss - alles angeblich, um ein besseres Benutzererlebnis zu bieten. Natürlich werden solche Dinge immer als gut für den Verbraucher oder den Bürger dargestellt, aber im Endeffekt ist der Benutzer einfach das Opfer.

Was die geplante Obsoleszenz angeht, so sind die grünen Abgeordneten mit ihrem Neuen Europäischen Bauhaus [6] vielleicht die Schlimmsten von allen. Ich spreche hier nicht vom deutschen Baustil, aber ich bin besorgt über den zunehmenden Druck auf den Bürger, seine noch einwandfreien Geräte auf den Schrottplatz zu werfen und sie durch energieeffizientere „grüne“ Versionen zu ersetzen. Abgesehen von den idiotischen Kosten scheint mir die Produktion und das Recycling von Haushaltsgeräten erhebliche CO<sub>2</sub>-Emissionen zu verursachen. Und jetzt bin ich per Gesetz gezwungen, meinen perfekt funktionierenden Gasheizkessel zu entsorgen und durch einen neuen zu ersetzen, nur um ein paar Prozent mehr Effizienz zu erreichen. Wenigstens wissen wir, wofür wir arbeiten müssen... ◀

220461-02



### WEBLINKS

[1] EDs aus Dubai: Die königlichen Lichter, die man nicht kaufen kann:

<https://hackaday.com/2021/01/17/leds-from-dubai-the-royal-lights-you-cant-buy/>

[2] Die Lampen, die man nicht haben darf. Entdeckung der Dubai-Lampen: <https://youtu.be/klaJqofCsu4>

[3] Herstellung von Dubai-Lampen aus 13-W-Philips-Leuchten mit einem einfachen Hack: <https://youtu.be/rXJq2vLJhLA>

[4] Der Epson-Drucker-Streit: geplante Obsoleszenz erhebt ihr hässliches Haupt: <https://elektor.link/epsonobsolescence>

[5] EEVblog 1462 - Warum Dymo-Etikettendrucker nerven: <https://youtu.be/xzSDJRC0F6c>

[6] Bericht über das neue europäische Bauhaus: [https://www.europarl.europa.eu/doceo/document/A-9-2022-0213\\_EN.html](https://www.europarl.europa.eu/doceo/document/A-9-2022-0213_EN.html)

# Reverse-Engineering eines LED-Displays mit Bluetooth Low Energy

Wie man ein BLE-Gerät mit einem Python-Skript steuert

Von Koen Vervloesem (Belgien)

Viele Geräte mit Bluetooth Low Energy lassen sich vom Benutzer mit einer eigenen „mobilen“ App steuern. Dabei kommt in der Regel ein benutzerdefiniertes Protokoll zum Einsatz, das nur von dem Gerät und der besagten App verstanden wird. Meist gibt es keine Spezifikation, die Sie nachlesen könnten, um Ihre eigene Implementierung zu erstellen. Doch jetzt kommt die gute Nachricht: Das Reverse-Engineering des BLE-Geräts ist der Weg, um es zu hacken und mit Ihrer eigenen Software zu verwenden.

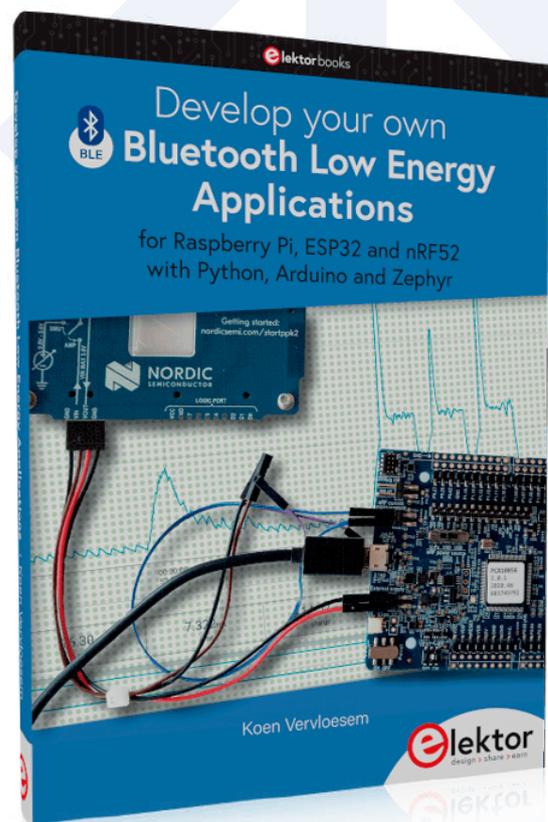
**Anmerkung der Redaktion:** Obwohl dieser Artikel auf einem Kapitel aus dem 257-seitigen Buch *Develop your own Bluetooth Low Energy Applications* (Elektor 2022) basiert, wurde der Inhalt vom Autor speziell für diese Elektor-Ausgabe angepasst, um Ihnen ein geschlossenes, vollständig reproduzierbares Projekt zu präsentieren.

In diesem Artikel möchte ich ein BLE-LED-Display, auch als LED-Badge bekannt, als Beispiel für diese anspruchsvolle und sehr lehrreiche Tätigkeit zurückentwickeln. Begleiten Sie mich und entdecken Sie, wie das Gerät funktioniert, indem Sie seine BLE-Dienste und Eigenschaften untersuchen. Begleiten Sie mich, wie ich die zugehörige mobile Anwendung dekompile und den BLE-Datenverkehr zwischen der Anwendung und dem Gerät abfange. Mein Ziel ist es, ein Python-

Skript zur Steuerung des LED-Displays zu erstellen, so dass ich die „offizielle“ mobile App nicht mehr benötige.

## Untersuchen des LED-Badges

Das LED-Badge von AliExpress (**Bild 1**) besitzt ein 11 x 55-LED-Pixel-Display, das in verschiedenen Farben erhältlich ist. Es unterstützt Bluetooth, wobei in den technischen Daten keine Bluetooth-Version angegeben ist.



Das Scannen des QR-Codes auf der Rückseite des Badges führt zu einem „HTTP 404“-Fehler. Bei Google Play fand ich eine App namens „Bluetooth LED Name Badge“, hergestellt von Shenzhen Lesun Electronics Co. Ltd (**Bild 2**). Diese App kann Text und Symbole an das Badge senden und verfügt über einige Einstellmöglichkeiten, darunter auch für die Bildlaufgeschwindigkeit.

Schalten wir das LED-Badge ein und verwenden die BLE-Scan-App „nRF Connect for Mobile“ [1], um zu sehen, was das Gerät anzeigt oder „advertised“, wie es genannt wird. Drücken Sie die untere Taste zweimal. Auf dem Display wird dann ein BLE-Symbol angezeigt. In nRF Connect werden Sie feststellen, dass das Gerät nun herstellere-spezifische Daten und zwei benutzerdefinierte BLE-Services anzeigt: 0xfee7 und 0xfee0 (**Bild 3**). Das Gerät trägt den Namen LSLED. Wenn man sich mit ihm verbindet, werden seine Services (Dienste) und die in **Tabelle 1** aufgeführten Characteristics (Merkmale) angezeigt.

**Tabelle 1. Merkmale von BLE-Badges**

Dienst	Merkmal	Eigenschaft
0xfee7	0xfec7	WRITE
0xfee7	0xfec8	INDICATE
0xfee7	0xfec9	READ
0xfee0	0xfee1	NOTIFY, READ, WRITE

Das Lesen des Merkmals „0xfec9“ in nRF Connect for Mobile liefert den gleichen Wert wie die herstellere-spezifischen Daten. Das Lesen von 0xfee1 gibt keine Daten zurück, und das Abonnieren seiner Benachrichtigungen liefert ebenfalls nichts. Die *Characteristic User Description* von 0xfee1 lautet lediglich „Daten“. Diese Methode liefert also nichts Nützliches.

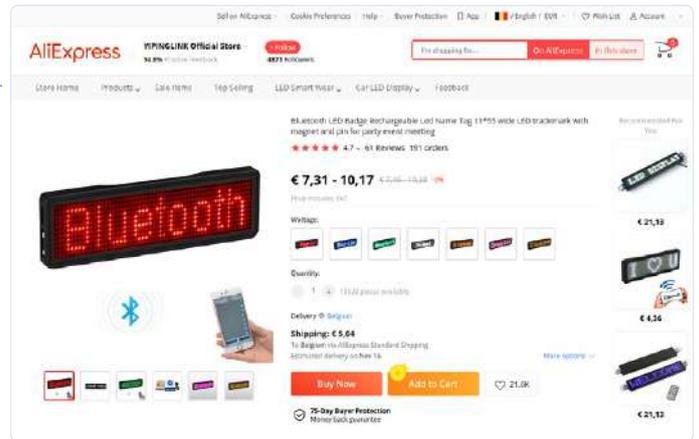
### Dekompilieren der mobilen Anwendung

Da die Android-App für das Bluetooth-LED-Badge offensichtlich in der Lage ist, mit dem LED-Badge zu kommunizieren, wollen wir versuchen herauszufinden, wie das die App anstellt. Dazu decompilieren wir die App und ergünden die Funktionsweise ihres Quellcodes.

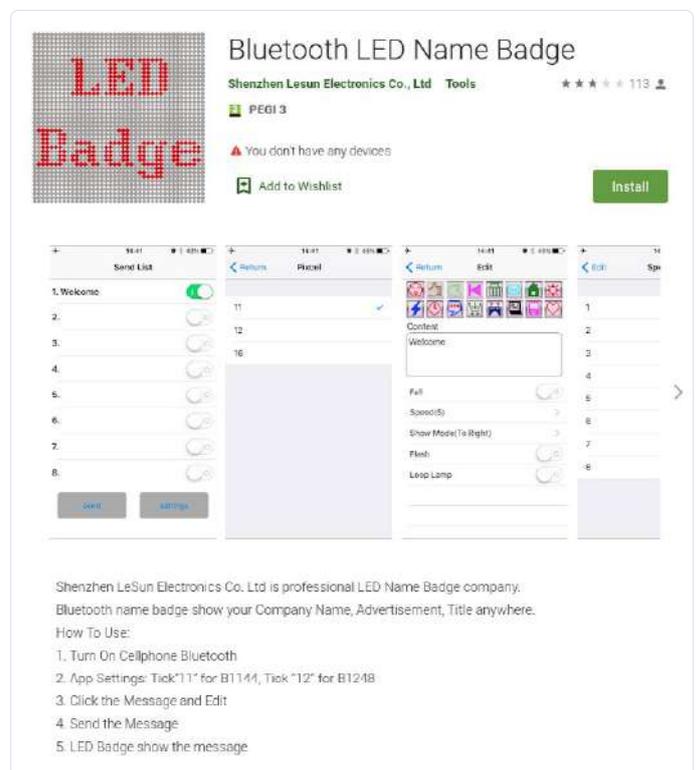
Zunächst laden wir die APK-Datei für die Android-App über die APK-Downloader-Webseite eines Drittanbieters wie [2] herunter. Fügen Sie dazu einfach die URL der App im Google Play Store [3] in das Suchfeld von APKPure ein. Diese APK-Datei enthält den Dalvik-Bytecode für Android, eine Art exe-Datei. Um zu verstehen, was die App tut, benötigen wir aber ihren Quellcode. Zwar hat der Entwickler den Quellcode in Dalvik kompiliert, aber wir können dies mit einem Decompiler wie jadx [4] rückgängig machen: Der Dalvik-Bytecode wird in Java-Quellcode zurückgewandelt.

Bei der neuesten jadx-Version für Windows [5] können wir die grafische Oberfläche durch einen Doppelklick auf die jadx-gui-Datei im bin-Verzeichnis starten. In anderen Betriebssystemen (Linux, MacOS) starten Sie die grafische Oberfläche, indem Sie *bin/jadx-gui* von der Kommandozeile aus ausführen.

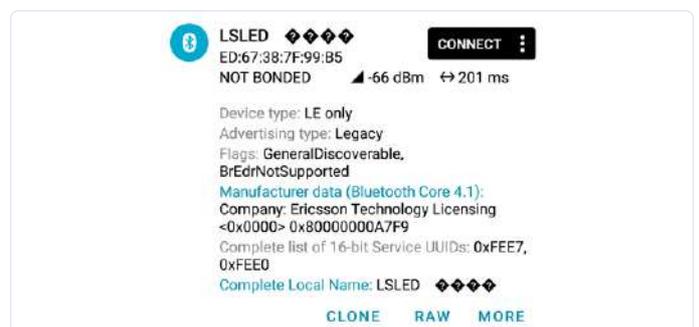
Öffnen Sie die heruntergeladene APK-Datei. Das Programm decompiliert nun die App und zeigt auf der linken Seite eine Baumstruktur der Pakete und Java-Dateien an. Nun beginnt die Suche. Wir haben bereits einige Hinweise: die UUIDs der Dienste und Merkmale, die



**Bild 1.** Dieses Bluetooth-LED-Badge scheint ein interessantes Gerät zu sein, das man „reverse-engineeren“ kann!



**Bild 2.** Die App „Bluetooth LED Name Badge“ aus dem Google Play Store kann Befehle an das LED-Badge senden.



**Bild 3.** Das Bluetooth-LED-Badge in nRF Connect for Mobile.



### Listing 1.

```
public static final String UUID_CHARACTERISTICS_WRITE = "fee1"
public static final String UUID_SERVICE = "fee0";
```



### Listing 2.

```
public static byte[] get64(List<SendContent> list, int i) {
    Iterator<SendContent> it = list.iterator();
    while (it.hasNext()) {
        Log.d("abcdef", "get64-----SendContent:" + it.next().toString());
    }
    byte[] bArr = new byte[64];
    bArr[0] = 119;
    bArr[1] = 97;
    bArr[2] = 110;
    bArr[3] = 103;
    bArr[4] = 0;
    bArr[5] = 0;
    bArr[6] = getFlash(list);
    bArr[7] = getMarquee(list);
    byte[] modeAndSpeed = getModeAndSpeed(list);
    for (int i2 = 0; i2 < 8; i2++) {
        bArr[i2 + 8] = modeAndSpeed[i2];
    }
    byte[] msgLength = getMsgLength(list, i);
    for (int i3 = 0; i3 < 16; i3++) {
        bArr[i3 + 16] = msgLength[i3];
    }
    bArr[32] = 0;
    bArr[33] = 0;
    bArr[34] = 0;
    bArr[35] = 0;
    bArr[36] = 0;
    bArr[37] = 0;
    byte[] date = getDate();
    for (int i4 = 0; i4 < 6; i4++) {
        bArr[i4 + 38] = date[i4];
    }
    for (int i5 = 0; i5 < 19; i5++) {
        bArr[i5 + 44] = 0;
    }
    bArr[63] = 0;
    return bArr;
}
```

in nRF Connect for Mobile gefunden wurden. Im Menü **Navigation / Text search** können Sie Suchbegriffe eingeben.

Das ist jedoch leichter gesagt als getan. Der Code erwähnt weder den Service 0xfee7 noch seine Characteristics! Ein anderer Dienst und seine Eigenschaften werden in der Klasse `com.yannis.ledcard.ble.BleDevice` erwähnt. Die relevanten Teile sind in **Listing 1** zu sehen.

Im Code können Sie nun nach den beiden Konstanten

`UUID_CHARACTERISTICS_WRITE`

`UUID_SERVICE`

suchen. Am einfachsten ist es wohl, mit der rechten Maustaste auf den Namen der Konstanten zu klicken und dann **Find usage** zu wählen.

Klicken Sie dann auf eine der gefundenen Instanzen, um die entsprechende Datei an dieser Stelle zu öffnen.

Auf der Suche nach weiteren Hinweisen im Code stieß ich auf einen vielversprechenden Code, der sich mit Bildern und Anzeigemodi in der Klasse

`com.yannis.ledcard.util.LedDataUtil` beschäftigt. Konkret habe ich diese Java-Methode in **Listing 2** gefunden, die eine Art Kopfzeile für Daten zu erstellen scheint.

Dies zeigt einen festen Header (sechs Bytes), einen Modus und eine Geschwindigkeit, eine Nachrichtenlänge und ein Datum. Es wird von einem `tSendHeader()` in der genannten Klasse aufgerufen.

Um zu verstehen, was die Anwendung tut, reichte es offensichtlich nicht aus, nur den Code näher zu untersuchen. Also begann ich, das LED-Badge mit der App zu steuern und den Datenverkehr zwischen den beiden Seiten abzufragen. Um ein besseres Verständnis zu erlangen, konnte ich nun das Wissen aus dem Quellcode der App auf den Live-Datenverkehr anwenden.

## BLE-Datenverkehr zwischen Badge und App

Mit Wireshark [6] lässt sich der laufende Bluetooth-Datenverkehr des Handys über die *Android Debug Bridge* abhören. Schließen Sie Ihr Handy mit einem USB-Kabel an Ihren Computer an und erlauben Sie die Debug-Verbindung auf Ihrem Handy. Starten Sie Wireshark. Dieses sollte *Android Bluetooth Btsnoop Net* als eine der verfügbaren Schnittstellen anzeigen. Nach einem Doppelklick auf die Schnittstelle sehen Sie, wie der Bluetooth-Verkehr live an Ihnen vorbeirollt! Eine noch bessere Option ist der *nRF Sniffer for Bluetooth LE plugin for Wireshark* [7], das einen nRF52840-Dongle als Hardwareteil des Sniffers verwendet.

Drücken Sie auf dem BLE-Badge zweimal die untere Taste, bis das Bluetooth-Symbol angezeigt wird. Wählen Sie dann das Gerät in Wireshark aus, um (nur) die Pakete zu und von diesem speziellen Gerät anzuzeigen.

Installieren Sie „Bluetooth LED Name Badge“ auf Ihrem Android-Handy und öffnen Sie die App. In Wireshark sehen Sie, dass Ihr Smartphone eine Scan-Anfrage stellt und das BLE-Badge mit seinem Gerätenamen in der Scan-Antwort reagiert. Wählen Sie den Gerätetyp in der App aus, für das 11 x 55 Pixel große LED-Badge ist dies der Typ 11. Tippen Sie auf **yes**.

Die App zeigt jetzt eine Liste der zu sendenden Nachrichten an. In der Standardkonfiguration wird eine Nachricht gesendet: **Welcome**. Tippen Sie auf **Send**. Die App stellt daraufhin eine Verbindung mit

dem LED-Badge her und zeigt die Willkommensnachricht auf dem Display an (**Bild 4**).

In Wireshark sehen wir ein CONNECT\_REQ-Paket, gefolgt von einer Anfrage nach Attributen. Dann folgen einige Write Requests und Responses. Einen besseren Überblick erhalten Sie, wenn Sie im Teil Bluetooth Attribute Protocol der Paketdetails mit der rechten Maustaste auf den Opcode Write Request klicken und dann **Apply as Filter / Selected** wählen. Alternativ dazu können Sie als Displayfilter `btatt.opcode == 0x12` eingeben.

In diesem Fall hat die App neun Write-Request-Pakete mit jeweils 16 Byte Daten gesendet, wie unten dargestellt:

```
77 61 6e 67 00 00 00 00 30 30 30 30 30 30 30 30
00 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 e5 0a 18 0c 37 25 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 c6 c6 c6 c6 d6 fe ee c6 82 00 00 00 00 00 7c
c6 fe c0 c6 7c 00 00 38 18 18 18 18 18 18 3c
00 00 00 00 00 7c c6 c0 c0 c6 7c 00 00 00 00 00
7c c6 c6 c6 c6 7c 00 00 00 00 00 ec fe d6 d6 d6
c6 00 00 00 00 00 7c c6 fe c0 c6 7c 00 00 00 00
```

Wenn Sie sich die Java-Methode `get64()` aus dem vorherigen Abschnitt ansehen, werden Sie am Anfang Teile des Headers erkennen: 77 61 6e 67 (hexadezimaler Äquivalent zu dezimal 119 97 110 103). Dann erhalten wir zweimal 00, und dann wieder zweimal 00, was die Werte für `getFlash()` beziehungsweise `getMarquee()` sind. Die nächsten acht Bytes, die den Modus und die Geschwindigkeit darstellen, sind alle 0x30. Die nächsten 16 Bytes stehen für die Nachrichtenlänge. Da die App eine Liste von acht Nachrichten anzeigt, könnte dies die Liste der Längen dieser Nachrichten sein. Die ersten beiden Bytes hier sind 00 07, was in der Tat die Anzahl der Zeichen des Textes **Welcome** ist. Die Hypothese ist, dass die nächsten 14 Bytes die Längen der nächsten sieben Nachrichten darstellen (in diesem Fall keine).

Dann gibt es sechs 00, und die nächsten sechs Bytes stehen für das Datum, das in diesem Fall e5 0a 18 0c 37 25 ist. In Dezimalzahlen umgerechnet ist das 229 10 24 12 55 37. Ich habe diese Anwendung am 24. Oktober 2021 um 12:55:37 Uhr ausgeführt. Der Monat, der Tag und die Uhrzeit waren korrekt. Der Header schließt mit zwanzigmal 00 ab. Danach folgen fünf Pakete mit 16 Bytes, die irgendwie die sieben Zeichen des **Welcome**-Textes repräsentieren müssen. Lassen Sie uns herausfinden, wie das geht, und mit einer einfacheren Nachricht beginnen. Öffnen Sie die App erneut und klicken Sie auf die erste Nachricht. Ändern Sie den Text in **W** und aktivieren Sie **Flash**. Kehren Sie zum Haupt-Screen zurück, klicken Sie auf die zweite Nachricht und fügen Sie den Text **e** hinzu. Stellen Sie für diese zweite Nachricht die Geschwindigkeit auf **8**, den Modus auf **right** und aktivieren Sie **Marquee**. Kehren Sie zum Hauptbildschirm zurück. Aktivieren Sie dann den Schieber neben der zweiten Nachricht und klicken Sie auf **Send**, während Sie sich die Bluetooth Attribute Protocol Pakete in Wireshark ansehen.

Die LED-Anzeige zeigt nun den Buchstaben **W**, der sich nach links bewegt und immer wieder aufblinkt. Danach wird der Buchstabe **e** angezeigt, der sich mit doppelter Geschwindigkeit nach rechts bewegt, und ein Rahmen mit sich bewegenden Pixeln um ihn herum. In diesem Fall hat die App sechs Write-Request-Pakete mit jeweils 16 Byte Daten gesendet.



Bild 4. Das Bluetooth-LED-Badge zeigt eine Willkommensnachricht an.

```
77 61 6e 67 00 00 01 02 30 71 30 30 30 30 30 30
00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 e5 0a 18 10 06 13 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 c6 c6 c6 c6 d6 fe ee c6 82 00 00 00 00 00 7c
c6 fe c0 c6 7c 00 00 00 00 00 00 00 00 00 00 00
```

Das erste Paket beginnt mit denselben sechs Bytes, aber der Wert von `getFlash()` ist 01 und der von `getMarquee()` 02. Das liegt daran, dass wir Flash für die erste Nachricht und Marquee für die zweite Nachricht aktiviert haben.

Das nächste Byte ist immer noch 0x30, weil wir nichts am Modus und der Geschwindigkeit der ersten Nachricht geändert haben. Aber das Byte danach lautet jetzt 0x71. Wir haben die Geschwindigkeit auf 8 und den Modus auf right geändert. Das linke Nibble dieses Bytes kodiert also offensichtlich die Geschwindigkeit (Geschwindigkeit 4 wird als 3 und Geschwindigkeit 8 als 7 kodiert) und das rechte Nibble den Modus. Das zweite Paket zeigt 00 01 als Länge der ersten Nachricht und dasselbe für die Länge der zweiten Nachricht. Dies bestätigt unsere Hypothese. Dann wird das Datum kodiert.

Es sind jetzt noch zwei Pakete übrig. Wir wandeln die hexadezimalen Werte in binäre Werte um und formatieren sie byteweise und der Reihe nach, dann teilen wir sie in elf Zeilen (ja, weil das Display elf Pixel hoch ist) wie folgt auf:

```
00000000
11000110
11000110
11000110
11000110
11010110
11111110
11101110
11000110
10000010
00000000

00000000
00000000
00000000
00000000
01111100
11000110
11111110
11000000
11000110
01111100
00000000
```





#### Listing 4.

```
"""Display an image on a BLE LED badge.
Copyright (c) 2022 Koen Vervloesem
SPDX-License-Identifier: MIT
"""
import asyncio
import sys
from PIL import Image
import bleak

WRITE_CHAR_UUID = "0000fee1-0000-1000-8000-00805f9b34fb"
COMMAND1 = bytes([0x77, 0x61, 0x6E, 0x67, 0x00, 0x00, 0x00, 0x00, 0x34, 0x30, 0x30, 0x30, 0x30, 0x30, 0x30, 0x30])
COMMAND2 = bytes([0x00, 0x07, *(14 * [0x00])])
COMMAND3 = bytes(16 * [0x00])
COMMAND4 = bytes(16 * [0x00])
def chunks(lst, n):
    """Yield successive n-sized chunks from lst."""
    for i in range(0, len(lst), n):
        yield lst[i : i + n]
def commands_for_image(image):
    """Return commands to show an image on the BLE LED badge."""
    image_bytes = bytearray()
    with Image.open(image) as im:
        px = im.load()
        for i in range(7): # 7x8 = 56 -> 7 bytes next to each other
            for row in range(11):
                row_byte = 0
                for column in range(8):
                    try:
                        pixel = int(
                            px[(i * 8) + column, row][3] / 255
                        )
                    except IndexError:
                        pass # Ignore the 56th pixel in a full row
                    row_byte = row_byte | (pixel << (7 - column))
                image_bytes.append(row_byte)
    # Fill the end with zeroes to have a multiple of 16 bytes
    image_bytes.extend(bytes(16 - len(image_bytes) % 16))
    # Split image bytes into 16-byte chunks
    return list(chunks(image_bytes, 16))
async def main(address, filename):
    """Connect to BLE LED badge and send commands to show an image."""
    try:
        async with bleak.BleakClient(address) as client:
            commands = [COMMAND1, COMMAND2, COMMAND3, COMMAND4]
            commands.extend(commands_for_image(filename))
            for command in commands:
                await client.write_gatt_char(WRITE_CHAR_UUID, command)
    except asyncio.exceptions.TimeoutError:
        print(f"Can't connect to device .")
    except bleak.exc.BleakError as e:
        print(f"Can't write to device : ")
if __name__ == "__main__":
    if len(sys.argv) == 3:
        address = sys.argv[1]
        filename = sys.argv[2]
        asyncio.run(main(address, filename))
    else:
        print(
            "Please specify the BLE MAC address and image filename."
        )
)
```

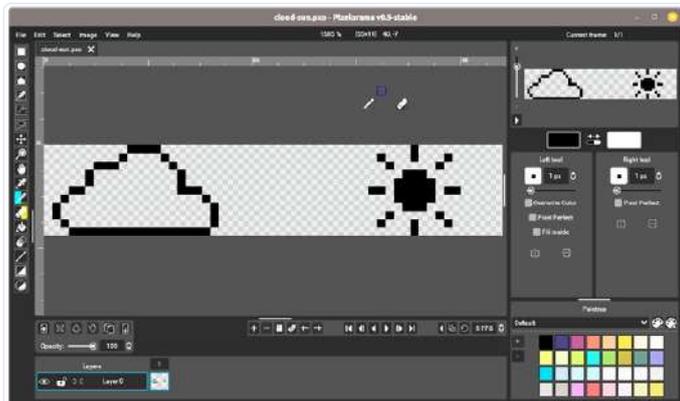


Bild 5. Pixelorama ist ein Open-Source 2D-Bildbearbeitungsprogramm.



Bild 6. Wenn Sie die entsprechenden BLE-Befehle senden, können Sie Ihre eigenen 11 x 55 Pixel großen Bilder auf dem LED-Badge anzeigen.

Der Hauptcode am unteren Ende prüft, ob Sie zwei Argumente in der Befehlszeile angegeben haben. Das erste Argument ist der Bluetooth-Adresse zugeordnet, das zweite dem Dateinamen.

Bevor Sie diesen Code ausführen, sollten Sie mit einem Pixel-Editor wie Pixelorama [10] ein Bild von genau 11 x 55 Pixeln vorbereiten. Füllen Sie das Pixel jeder LED, die auf dem Display leuchten soll. **Bild 5** zeigt ein Beispiel für ein Wolken- und Sonnenbild, das ich in Pixelorama erstellt habe.

Exportieren Sie das Bild als PNG-Datei. Versetzen Sie nun das LED-Badge in den Discovery-Modus, indem Sie zweimal auf die untere Taste drücken, so dass das Bluetooth-Symbol erscheint. Führen Sie das Python-Skript mit zwei Argumenten aus, zuerst die Bluetooth-Adresse des Badges und dann den Dateinamen des Bildes:

```
$ python3 display_led_badge.py ED:67:38:7F:99:B5 cloud-sun.png
```

## Fazit

Das LED-Badge sollte nun Ihr Bild anzeigen (**Bild 6**). Sie haben das BLE-Badge erfolgreich zurückentwickelt und sind nun in der Lage, das Gerät mit einem eigenen Code zu verwenden. Sicherlich kann mein vorgeschlagenes Skript noch verbessert werden, insbesondere um es allgemeiner zugänglich zu machen. Ich hoffe, dies inspiriert Sie dazu, die gleiche Art von Reverse Engineering auf andere BLE-Geräte anzuwenden. ◀

(220439-02)RG

## Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an den Autor unter [koen@vervoesem.eu](mailto:koen@vervoesem.eu) oder an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



### PASSENDE PRODUKTE

- **K. Vervoesem, Develop your own Bluetooth Low Energy Applications, Elektor 2022**  
Buch SKU 20200: [www.elektor.de/20200](http://www.elektor.de/20200)

**Hinweis: Dieses Buch wird mit einem kostenlosen nRF52840-USB-Dongle geliefert!**

- **K. Vervoesem, Develop your own Bluetooth Low Energy Applications, Elektor 2022**  
E-Buch SKU 20201: [www.elektor.de/20201](http://www.elektor.de/20201)

## WEBLINKS

- [1] nRF Connect for Mobile App: <https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-mobile>
- [2] Die Datei APK: <https://apkpure.com>
- [3] Die App im Google Play Store: <https://play.google.com/store/apps/details?id=com.yannis.ledcard>
- [4] jadx-Decompiler: <https://github.com/skylot/jadx>
- [5] Neueste jadx-Version: <https://github.com/skylot/jadx/releases>
- [6] Wireshark: <https://www.wireshark.org>
- [7] nRF Sniffer für Bluetooth LE : <https://www.nordicsemi.com/Products/Development-tools/nrf-sniffer-for-bluetooth-le>
- [8] Bleak : <https://bleak.readthedocs.io>
- [9] Die Pillow-Bibliothek: <https://pillow.readthedocs.io>
- [10] Pixelorama: <https://orama-interactive.itch.io/pixelorama>
- [11] Buch/Infoseite: <https://www.elektor.de/develop-your-own-bluetooth-low-energy-applications>
- [12] Code auf GitHub: <https://github.com/koenvervoesem/bluetooth-low-energy-applications>

# MakePython ESP32 Development Kit

Alles in einer Box

Von Tam Hanna (Slowakei)

Moderne Mikrocontroller wie der ESP32 sind so leistungsfähig, dass sie in MicroPython programmiert werden können - dank der mächtigen Bibliotheken kommt man hier schnell zu einem fertigen Projekt. Mit dem MakePython ESP32 Development Kit - es handelt sich dabei einerseits um ein Lehrbuch und andererseits um ein Hardware-Kit - schickt der bekannte Elektor-Autor Dogan Ibrahim nun ein Starterpaket ins Rennen, das die Arbeit mit MicroPython anhand realer Beispiele illustriert.



Dass MicroPython nicht der Weg zur „maximalen“ Programm-beziehungswise Software-Effizienz darstellt, muss man nicht wirklich diskutieren. Andererseits gilt, dass moderne Mikrocontroller wie ein ESP32 von der Leistungsfähigkeit her mit einem 486er mehr als mithalten - insbesondere in Kleinserien kann es deshalb vernünftiger sein, durch Nutzung von Hochsprachen einen Tausch von Programmier-Aufwand gegen Geschwindigkeit durchzuführen.

## Das Buch

Beginnen wir mit dem - in Englisch vorliegenden - Lehrbuch: Interessant ist die (gelungene) Aufmachung, die eine Fake-Spiralbindung vortäuscht.

Der eigentliche didaktische Aufbau beginnt dann mit einer Kurz-Erklärung des ESP32 als Ganzes: Als Elektor-Leser, der mit Mikrocontrollern schon umfangreiche oder zumindest grundlegende Erfahrung mitbringt, bekommen Sie so schnell einen Überblick über die verschiedenen von Espressif im Controller zur Verfügung gestellten Peripheriegeräte. Haben Sie aber mit Mikrocontrollern noch überhaupt keine Erfahrung, dürften die Ausführungen -

zumindest stellenweise - zum bequemen Verständnis zu kurz sein. Die Installation der IDEs – Ibrahim stellt uPyCraft und Thonny gleichermaßen vor, arbeitet später aber fast nur noch mit Thonny - wird unter Windows detailliert erklärt, Linux kommt nicht so sehr zu Sprache. Danach folgt noch ein Kapitel, das die Ausführung einiger kleiner Python-Snippets unter Nutzung von Thonny illustriert - wer keine Kenntnisse der Python-Syntax mitbringt, kann sich an dieser Stelle nicht wirklich einarbeiten. Andererseits reichen die Ausführungen dicke aus, um die Grundlagen der Arbeit mit MicroPython und der IDE zu verstehen.

Die eigentliche Vorstellung der im Werk enthaltenen 46 Projekte erfolgt dann im klassischen Dogan-Ibrahim-Stil: Im ersten Schritt präsentiert der Professor immer die „zu erledigende Gefechtsaufgabe“, um danach Code und Ausführungen zum Hardware-Aufbau zu präsentieren. Dass die Listings immer einen standardisierten Header mit gut zehn Zeilen Umfang mitbringen, „nervt“ insbesondere Freunde der kompakten Kodierung ein wenig. Andererseits muss man Ibrahim in diesem Werk zugutekommen lassen, dass die Beispiele zwar nicht kompliziert sind, aber doch die wesentlichen Aspekte der Arbeit mit MicroPython treffend illustrieren.



Bild 1. Eine Plastikverpackung schützt die Komponenten des Kits zuverlässig vor Beschädigung.

Die verzweifelte Suche danach, wie man bestimmte Peripheriegeräte in Betrieb nimmt, haben sie nach der Lektüre mit Sicherheit nicht mehr zu befürchten. Sehr lobenswert ist nach Ansicht des Rezensenten, dass Ibrahim erklärt, wie man ein „zerschossenes“ Board mit einer neuen MicroPython-Firmware ausstattet.

### Das Board im Blick

Apropos Evaluationsboard: Das Lehrbuch ist im Zusammenspiel mit dem in **Bild 1** gezeigten Kit erhältlich, das ob der durchaus robusten Plastikverpackung auch zur Mitnahme in den Urlaub oder zur Verhütung von Langeweile bei einer Geschäftsreise taugt. Doch damit zur eigentlichen Hardware: Die praktische Lehr-Erfahrung des Autors zeigt, dass insbesondere embedded-fremde Entwickler leichter lernen, wenn die Entwicklungs-Plattform ein (idealerweise vollgrafisches) Display zur Verfügung stellt.

Elektor umgeht dieses Problem insofern geschickt, als es die in **Bild 2** gezeigte rote Platine ins Rennen geschickt - der schwarze Bereich ist kein schwarzes Loch, sondern eines der weit verbreiteten SSD1306-OLEDs mit einer Auflösung von 128x64 Pixeln.

An dieser Stelle verbirgt sich der nach Ansicht des Autors größte Kritikpunkt: Die 2,54-mm-Header sind nicht eingelötet. Dies ist insofern schade, als dass eine Person, die das Kit direkt in einem Store kauft und auf Urlaub mitnimmt, an dieser Stelle gelackmeiert ist – zumindest wenn man keinen Lötkolben zur Installation der Header zur Hand hat und der Handyshop nebenan nicht mithilft. Sonst bietet das Board keinen Anlass zur Kritik: Die verwendete ESP32-Variante ist eines der größeren Modelle und hat ausreichend Speicher, das MicroUSB-Interface sorgt dafür, dass „Kommandogerät-Kabel“ schnell von jedem älteren Handy entlehnt werden können. Außerdem findet sich im Kit auch ein ganz kurzes Kabel brauchbarer Qualität.

### Eine Demo

Während sich das Lehrbuch vor allem an den Bedürfnissen von unter Windows arbeitenden Entwicklern orientiert, wird der Autor in den folgenden Schritten - schon aus Gründen der Bequemlichkeit - auf Ubuntu 20.04 LTS setzen. Die unter <https://github.com/DFRobot/uPyCraft> bereitstehende IDE ist dabei eine eher schlechte Wahl, weil ihre Ausführung in allen Ubuntu-Versionen neuer als 16.04 mit einer nach dem Schema `ImportError: /tmp/_MEIOhQKhz/libz.so.1: version `ZLIB_1.2.9' not found (required by /usr/lib/x86_64-linux-gnu/libpng16.so.16)` aufgebauten Fehlermeldung scheitert.

„Schöner“ ist Thonny, das sich durch Eingabe von

```
bash <(wget -O - https://thonny.org/installer-for-linux)
```



Bild 2. Diese Platine ermöglicht das „schnelle Starten“ mit MicroPython.

automatisiert installieren lässt - der Start erfolgt danach durch die Eingabe des Kommandos `/home/tamhan/apps/thonny/bin/thonny`, für die Deinstallation greifen wir auf `/home/tamhan/apps/thonny/bin/uninstall` zurück. Im Rahmen des ersten Starts ist dann nur noch ein Sprachauswahl-Assistent abzunicken.

Im nächsten Schritt bietet es sich auch schon an, das Evaluationsboard mit einer Workstation zu verbinden. An dieser Stelle denkt Elektor mit und liefert das Board mit einer vorkonfigurierten MicroPython-Runtime aus. Besonders empfehlenswert ist in diesem Zusammenhang, dass die Platine auch gleich das Display einschaltet, was eine erste Funktionskontrolle erleichtert. Als Konverter kommt dabei übrigens ein Chip vom Typ Silicon Labs CP210x UART-Bridge zum Einsatz.

An dieser Stelle können Sie zu Thonny (**Bild 3**) wechseln und müssen darauf achten, im unten rechts eingeblendeten Auswahl-Menü die Python-Version anzuklicken und sich danach für die Version `MicroPython (ESP32)` entscheiden. Thonny geht daraufhin automatisiert auf die Suche nach ESP32-Boards; im Fall der hier verwendeten Platine wurde das Board automatisch erkannt. Über die Frage, ob Sie Thonny mit Superuser-Rechten ausführen müssen, lässt sich diskutieren - da das Benutzerkonto des Autors Mitglied der `plugdev`-Gruppe ist, konnte er Thonny auch in seinem normalen User-Account ausführen und trotzdem mit dem ESP32 interagieren.

### Ein erster Willkommensgruß!

Als nächste Amtshandlung wollen wir dazu übergehen, das OLED-Display des Geräts in Betrieb zu nehmen. Hierzu öffnen wir die URL <https://www.makerfabs.com/makepython-esp32-starter-kit.html>, und laden das Archiv `MakePython ESP32 Lessons Source Code` herunter - die Nutzung von `.rar` ist unter Linux ein wenig ärgerlich. Extrahieren Sie das Archiv jedenfalls in einen bequem zugänglichen Ort. Wir benötigen für die folgenden Schritte vor allem die Datei `ssd1306.py`, die den OLED-Treibercode zur Verfügung stellt. Klicken Sie im ersten Schritt auf `File Open` und entscheiden Sie sich danach für die Option `This Computer`. Im nächsten Schritt navigieren Sie zur Datei, um sie im Editor zu laden.

Sodann entscheiden wir uns für `File Save as` und dann für die Option `MicroPython Board`.

Sofern Thonny sich an dieser Stelle über das „Beschäftigtsein“ des Backends aufregt, können Sie in der Toolbar auf das rote Stopp-Symbol klicken, um einen Halt zu befehlen. Im nächsten Schritt speichern Sie die Datei unter dem bekannten Namen ins Dateisystem - zumindest dann, wenn Sie nicht die Original-Firmware unverändert belassen haben. In diesem Fall ist die Datei nämlich schon am Platz.

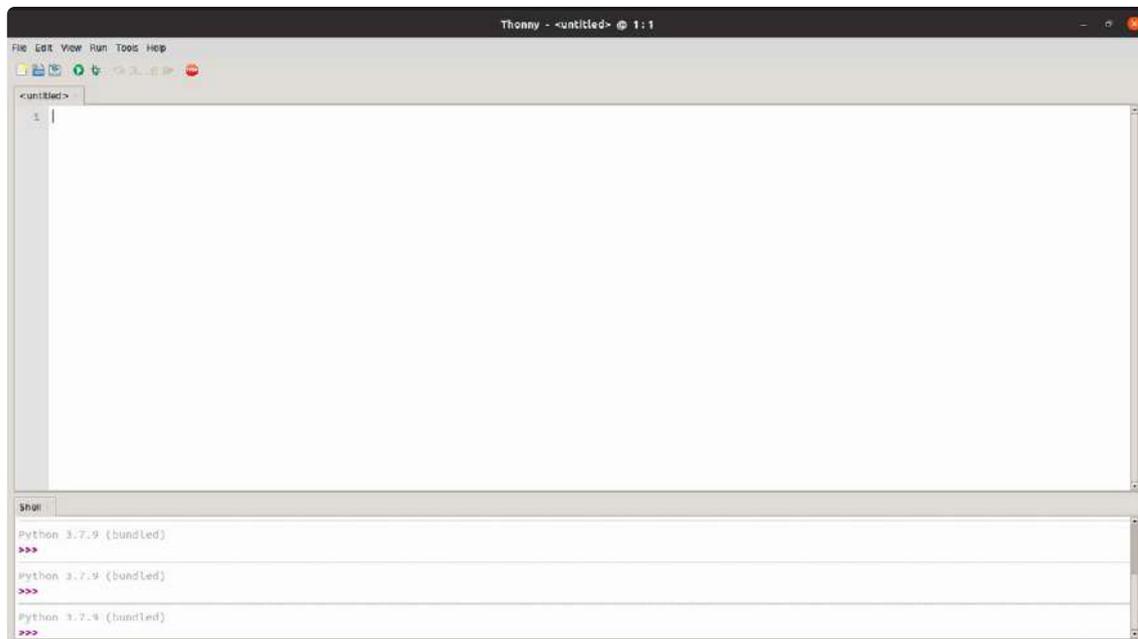


Bild 3. Screenshot von Thonny.

Als erste Amtshandlung importieren wir - unter MicroPython ist dies üblich - einige Bibliotheken, und legen auch Konstanten mit zusätzlichen Informationen über die Topologie des Displays fest:

```
import machine
import ssd1306
import time
WIDTH = const(128)
HEIGHT = const(64)
```

Für die eigentliche Kommunikation mit dem per I<sup>2</sup>C angeschlossenen Display - dies ist unüblich, meist trifft man hier auf SPI - setzen wir dann auf eine Instanz der Software-I<sup>2</sup>C-Klasse:

```
pscl = machine.Pin(5, machine.Pin.OUT)
psda = machine.Pin(4, machine.Pin.OUT)
i2c = machine.I2C(scl=pscl, sda=psda)
oled = ssd1306.SSD1306_I2C(WIDTH, HEIGHT, i2c)
```

Zu guter Letzt müssen wir dann nur noch nach folgendem Schema einen Text in Richtung des Displays schreiben:

```
while True:
    oled.fill(0)
    oled.text('Hello World!', 10, 0)
    oled.show()
    time.sleep(1)
```

An dieser Stelle können Sie in der IDE auch schon auf das Play-Symbol klicken, um sich am Aufscheinen des Texts auf dem Display des Evaluationsboards zu erfreuen.

## Einrichtung von MQTT

Als nächste Aufgabe wollen wir ein wenig über den gewöhnlichen, im Kit enthaltenen Projekt-Schatz hinausgehen, und eine fortgeschrittene Aufgabe realisieren - dies zeigt unter anderem auch, dass das Kit für höhere Aufgaben geeignet ist. Spezifischerweise wollen wir das universell bekannte MQTT-Protokoll verwenden. Besonders interessant wird die Aufgabe dadurch, dass wir hier mit einer brandaktuellen Version des MQTT-Brokers Mosquitto arbeiten wollen

- unter der URL <https://github.com/micropython/micropython-lib/issues/445> findet sich eine (lesenswerte) Beschreibung der „Probleme“, die die aus Sicherheitsgründen immer strikter eingestellten Parser im Zusammenspiel mit den MicroPython-Treibern bekommen.

Apropos MikroPython-Treiber: Es gibt für MicroPython mittlerweile sogar zwei konkurrierende Implementierungen von MQTT. Wir wollen hier allerdings auf die unter <https://github.com/micropython/micropython-lib/tree/master/micropython/umqtt.simple> stehende einfachere Variante setzen - laden Sie die Datei <https://github.com/micropython/micropython-lib/blob/master/micropython/umqtt.simple/umqtt/simple.py> herunter, und speichern diese danach unter dem Namen `simple.py` auf dem Prozessrechner.

Im nächsten Schritt müssen wir nach folgendem Schema die WLAN-Transmitter des Boards in den Station-Modus versetzen:

```
. . .
wlan=network.WLAN(network.STA_IF)
```

Für den eigentlichen Verbindungsaufbau setzen wir dann auf eine nach dem folgenden Schema aufgebaute Methode - sie stammt im Großen und Ganzen aus dem von MakerFabs zur Verfügung gestellten Projektskelett und realisiert einen Countdown-Timer samt Verbindungsaufbau. Achten Sie natürlich darauf, die an `wlan.connect` übergebenen Strings an ihre Drahtlosnetzwerk-Situation anzupassen:

```
def connectWiFi():
    i=0
    wlan.active(True)
    wlan.disconnect()
    wlan.connect("ssid", "pass")
    while(wlan.ifconfig()[0]!='0.0.0.0'):
        i=i+1
        oled.fill(0)
        oled.text('connecting WiFi', 0, 16)
        oled.text('Countdown: '+str(20-i)+'s', 0, 48)
        oled.show()
        time.sleep(1)
    if(i>20):
        break
```



### Listing 1.

```
def connectWiFi():
    i=0
    wlan.active(True)
    wlan.disconnect()
    wlan.connect("ssid", "pass")
    while(wlan.ifconfig()[0]!='0.0.0.0'):
        i=i+1
        oled.fill(0)
        oled.text('connecting WiFi',0,16)
        oled.text('Countdown:'+str(20-i)+'s',0,48)
        oled.show()
        time.sleep(1)
        if(i>20):
            break
    oled.fill(0)
    oled.text('Makerfabs',25,0)
    oled.text('MakePython ESP32',0,32)
    if(i<20):
        oled.text('WIFI connected',0,16)
    else:
        oled.text('NOT connected!',0,16)
    oled.show()
    time.sleep(3)
    return True
```

```
oled.fill(0)
oled.text('Makerfabs',25,0)
oled.text('MakePython ESP32',0,32)
if(i<20):
    oled.text('WIFI connected',0,16)
else:
    oled.text('NOT connected!',0,16)
oled.show()
time.sleep(3)
return True
```

Der eigentliche Test-Verbindungsaufbau könnte dann einfacher nicht sein:

```
connectWiFi()
while True:
    time.sleep(1)
```

Im nächsten Schritt wollen wir uns einen Mosquitto-Server zur Verfügung stellen: Mosquitto ist zwar nicht der schnellste MQTT-Broker, aber quelloffen, kostenlos und im Bereich der MQTT-Implementierung sehr streng.

Ob seiner hohen Verbreitung ist außerdem sichergestellt, dass es im Docker-Hub ein mehr oder weniger schlüsselfertiges Image gibt, dass die Bereitstellung des Servers ohne tiefere Konfigurationen des Host-Computers ermöglicht.

Für einen ersten Anlauf reicht es aus, folgenden Befehl einzugeben:

```
docker run -it -p 1883:1883 -p 9001:9001 -v mosquitto.conf:/mosquitto/config/mosquitto.conf eclipse-mosquitto
```

Wichtig sind hier erstens die Parameter `-p 1883:1883` und `-p 9001:9001` - jeder Docker-Container wird von der Runtime ja mit



einem vollen Komplement an TCP/IP-Ports ausgestattet. Jeder der Parameter verbindet dann einen dieser Ports mit der Netzwerkkarte des Host-Computers. Über den Parameter `-v mosquitto.conf:/mosquitto/config/mosquitto.conf` integrieren wir außerdem eine Konfigurationsdatei.

An dieser Stelle bietet es sich an, bei bestehender Internetverbindung einen ersten Download-Versuch durchzuführen. Docker wird normalerweise Verbindung zum Hub aufnehmen und die benötigten Komponenten herunterladen, um danach mit der folgenden Fehlermeldung zu verenden:

```
docker: Error response from daemon: source /var/lib/docker/aufs/mnt/a22b9e557c37d99eb71f17e7bc6d38df6e7677d09225376d416612adf0977ccd/mosquitto/config/mosquitto.conf is not directory.
```

Ursache des Fehlers ist, dass es in der Host-Workstation noch keine Datei mit dem Namen `mosquitto.conf` gibt, die wir dem Container zur Verfügung stellen könnten.

Zur Lösung reicht es aus, auf der Kommandozeile im Arbeitsverzeichnis mit `gedit` ein neues File anzulegen:

```
(base) tamhan@tamhan-thinkpad:~$ gedit mosquitto.conf
```

Der Umstieg von Mosquitto Version 1 auf Mosquitto Version 2 ging mit massiven Verschärfungen im Bereich der Sicherheits-Konfiguration einher. Während ein mit Default-Parameter gestarteter Mosquitto 1.X-Serveranfragen von beliebigen Clients entgegennahm, verweigert Mosquitto 2 dies. Wenn Sie allerdings die folgenden Passagen in der Konfigurationsdatei platzieren, funktioniert alles wieder wie gewohnt (und unsicher):

```
listener 1883
allow_anonymous true
```

Die eigentliche Ausführung lässt sich dann nach folgendem Schema befehligen - beachten Sie, dass der Parameter `-v` immer einen voll qualifizierten Pfad verlangt, weshalb wir das Kommando `pwd` aufrufen und die Ergebnisse durch Shell-Magie „verwursteln“. `pwd` steht dabei übrigens für Print Working Directory, **Bild 4** zeigt das Verhalten:

```
(base) tamhan@tamhan-thinkpad:~$ docker run -it -p 1883:1883 -p 9001:9001 -v $(pwd)/mosquitto.conf:/mosquitto/config/mosquitto.conf eclipse-mosquitto
. . .
```

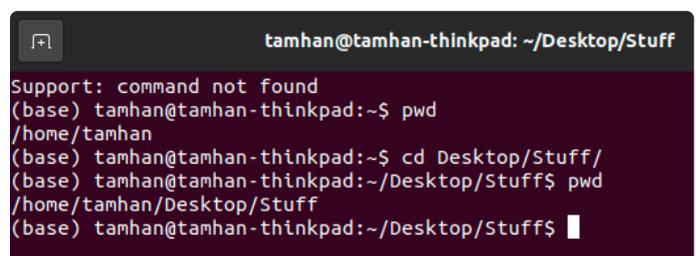


Bild 4. Der `pwd`-Befehl liefert das aktuelle Arbeitsverzeichnis der Shell.



Sofern der Docker-Container seinen erfolgreichen Start durch Rückgabe von `1658673183: mosquitto version 2.0.14 running quitted`, sind wir an dieser Stelle gefechtsbereit.

## Inbetriebnahme der MQTT-Software am ESP32

Als Erstes benötigen wir auch für die Arbeit mit MQTT einige Konstanten - achten Sie darauf, die im Server-String übergebene IP-Adresse an Ihre lokale Betriebssystemsituation anzupassen:

```
SERVER = "192.168.178.146"
CLIENT_ID = ubinascii.hexlify(machine.unique_id())
TOPIC = b"led"
state = 0
```

Außerdem benötigen wir eine Callback-Funktion, die der MQTT-Treiber immer beim Eingehen einer Nachricht aufrufen wird:

```
def sub_cb(topic, msg):
    global state
    print((topic, msg))
    . . .
```

Die eigentliche Einrichtung der MQTT-Verbindung ist dann vergleichsweise simpel:

```
connectWiFi()
c = MQTTClient(CLIENT_ID, SERVER, keepalive=30)
c.set_callback(sub_cb)
c.connect()
c.subscribe(TOPIC)
print("Connected to %s, subscribed to %s topic" % (SERVER,
TOPIC))

while True:
    c.wait_msg()
```

Der Aufruf von `c.subscribe` sorgt dafür, dass der MQTT-Treiber dem Server mitteilt, an welchen Nachrichten-Kanälen aktuell Interesse besteht. Wichtig ist außerdem noch das periodische Aufrufen der Methode `c.wait_msg()`, um dem MQTT-Server Rechenleistung zur Abarbeitung der eingehenden und der zu sendenden Informationen zur Verfügung zu stellen. In der Kommandozeile beziehungsweise im Ausgabefenster dürfen Sie sich dann an der Ausgabe `Connected to 192.168.178.146, subscribed to b'led' topic` erfreuen.

Im nächsten Schritt möchten wir noch eine der mitgelieferten Leuchtdioden samt dem beigelegten Widerstand mit dem GPIO-Pin 12 verbinden - der Autor geht davon aus, dass ausreichend Elektronik-Kenntnisse beim Leser verfügbar sind. Die Initialisierung des GPIO-Ports ist dann ebenfalls gewöhnlicher ESP32-Code:

```
led = Pin(12, Pin.OUT, value=1)
```

Für die Verarbeitung der eingehenden Nachrichten müssen wir den Callback dann insofern erweitern, als dass wir den im Wert `topic` angelieferten String mit den für die verschiedenen Befehle vorgesehenen Konstanten verdrahten:

```
def sub_cb(topic, msg):
    global state
    print((topic, msg))
    if msg == b"on":
        led.value(1)
        state = 1
    elif msg == b"off":
        led.value(0)
        state = 0
    elif msg == b"toggle":
        led.value(state)
```

Für das eigentliche Testen wollen wir dann auf die Kommandozeilen-Utility `mosquitto_pub` setzen: Es handelt sich dabei um ein unter Linux zur Verfügung stehendes Werkzeug, das das „direkte“ Absetzen von Kommandos in Richtung eines MQTT-Servers erlaubt. Zum Ein- und Ausschalten der LED dienen dann die folgenden Kommandos:

```
(base) tamhan@tamhan-thinkpad:~$ mosquitto_pub -t led -m 'on'
(base) tamhan@tamhan-thinkpad:~$ mosquitto_pub -t led -m 'off'
```

Sofern ESP 32 und Server im gleichen Netzwerk sind, können Sie ab sofort die Leuchtdiode ein und ausschalten.

## Fazit und Ausblick!

Das MakePython ESP32 Development Kit ist ein faszinierend-kompakter Werkzeugkasten, der Python-Entwicklern und/oder Elektronikern schnell die Vereinigung der beiden Welten ermöglicht. Insbesondere weil alles so schön schlüsselfertig verpackt ist, ist dies eine Box, dem der Rezensent gerne eine Empfehlung ausspricht. ◀

220429-02



### Passendes Produkt

➤ **MakePython ESP32 Development Kit (SKU 20137)**  
[www.elektor.de/20137](http://www.elektor.de/20137)

## WEBLINKS

[1] uPyCraft-IDE: <https://github.com/DFRobot/uPyCraft>

[2] umqtt.simple und mosquitto 2.0.12: <https://github.com/micropython/micropython-lib/issues/445>

[3] umqtt.simple - einfacher MQTT-Client für MicroPython:

<https://github.com/micropython/micropython-lib/tree/master/micropython/umqtt.simple>

## THD-Messung mit Oszilloskop und FFT

Den Klirrfaktor einfach berechnen

Von Sebastian Westerhold (AI5GW) (Deutschland)

Die Total Harmonic Distortion, kurz THD, ist ein wichtiges Maß für den Oberwellengehalt von Signalen in Stromversorgungsnetzen und bei elektronischen Baugruppen wie beispielsweise Audioverstärkern. Üblicherweise sind zur messtechnischen Erfassung spezielle Messgeräte notwendig. Wer allerdings ein Digitaloszilloskop mit FFT-Funktion zur Verfügung hat, kann mit etwas Mathematik auch damit die THD bestimmen.

Vorab ein paar Begrifflichkeiten: Harmonische beschreiben durch ihre Ordnungszahl sowohl die Grundwelle als auch die Oberwellen. Oberwellen beschreiben alle ganzzahligen Vielfachen der Grundwelle, beginnend mit der zweifachen Frequenz der Grundwelle. Die erste Harmonische entspricht der Grundwelle, die zweite Harmonische entspricht dem ersten Oberton.

Es gibt zwei wichtige Vertreter der THD-Familie, die  $THD_R$  und  $THD_F$ . Für beide gilt: Je kleiner die THD, desto oberwellenärmer ist das gemessene Signal. Je größer der Oberwellenanteil, umso größer ist die THD. Da Oberwellen typischerweise als Effekt von Verzerrungen auftreten, handelt es quasi sich um eine Messgröße zur Quantisierung von nicht-linearen Verzerrungen.

In diesem Artikel wird der Begriff „Oberwellen“ verwendet, wenn die Vielfachen der Grundfrequenz in ihrer Gesamtheit gemeint sind. Der Begriff „Harmonische“ wird dann verwendet, wenn entweder die Gesamtheit aller Frequenzbestandteile (inklusive der Grundwelle) oder ein spezifisches, durch eine Ordnungszahl gekennzeichnetes Vielfaches der Grundwelle gemeint ist. Beispiel: 3. Harmonische = dreifache Frequenz der Grundwelle.

Die  $THD_F$  bezeichnet das Verhältnis von allen Oberwellenanteilen zur Grundwelle. Dieses Maß findet zum Beispiel bei der messtechnischen

Erfassung von Oberwellen in Stromversorgungsnetzen Anwendung. Die  $THD_R$  bezeichnet hingegen das Verhältnis von allen Oberwellenanteilen zum Gesamtsignal. Diese Messgröße ist vor allem in der Tontechnik sehr beliebt und – zumindest in Deutschland – auch unter dem Namen Klirrfaktor bekannt.

Die Unterscheidung zwischen den beiden Messgrößen ist sehr wichtig, da  $THD_F$  und  $THD_R$  zwar eng miteinander verwandt sind, jedoch unterschiedliche Ergebnisse erzielen. In der Literatur und in Datenblättern werden die beiden Werte häufig miteinander verwechselt [1]. Stellenweise findet man auch nur die Angabe „THD“ ohne weitere Definition des verwendeten Verfahrens, beispielsweise im Datenblatt des LM386 von Texas Instruments.

In der Praxis lässt sich die  $THD_F$  relativ einfach aus der Summe der Leistungspegeldifferenzen aller Oberwellen in Relation zur Grundwelle errechnen. Daraus lässt sich dann wiederum die  $THD_R$  berechnen.

$$THD_F = 100 \cdot \sqrt{\sum_{n=2}^{\infty} 10^{\left(\frac{P_n}{10}\right)}} = 100 \cdot \sqrt{10^{\left(\frac{P_2}{10}\right)} + 10^{\left(\frac{P_3}{10}\right)} + 10^{\left(\frac{P_4}{10}\right)} \dots}$$

**Formel 1:**  $THD_F$  (in %),  $n$  = Ordnungszahl der Harmonischen,  $P$  = Leistungspegeldifferenz zur Grundwelle in dBc.

Die dem Prinzip zugrundeliegende Formel mag auf den ersten Anblick erst einmal abschreckend wirken. Vereinfacht gesagt entspricht die  $THD_F$  der Wurzel aus der Summe aller aus den relativen Leistungspegeldifferenzen errechneter Spannungsverhältnisse unendlich vieler Oberwellen relativ zur Grundwelle. Dieser Satz mag auf den ersten Blick genau so abschreckend wirken wie die Formel selbst. So kompliziert, wie es wirkt, ist die Sache jedoch nicht.

Die so errechnete  $THD_F$  lässt sich dann mit der folgenden Formel in den Klirrfaktor umrechnen:

$$THD_R = \frac{THD_F}{\sqrt{1 - THD_F^2}}$$

**Formel 2:**  $THD_R$  (in %),  $THD_F$  (in %).

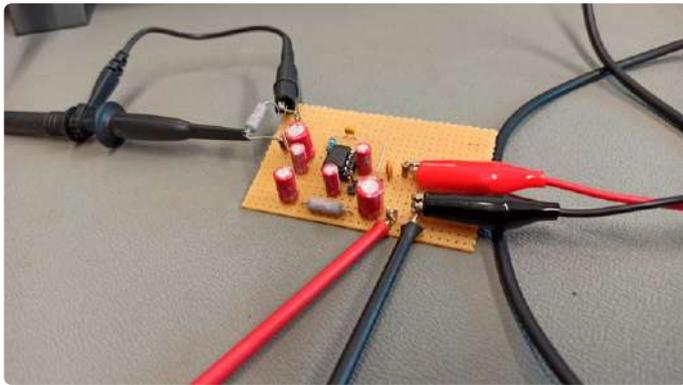
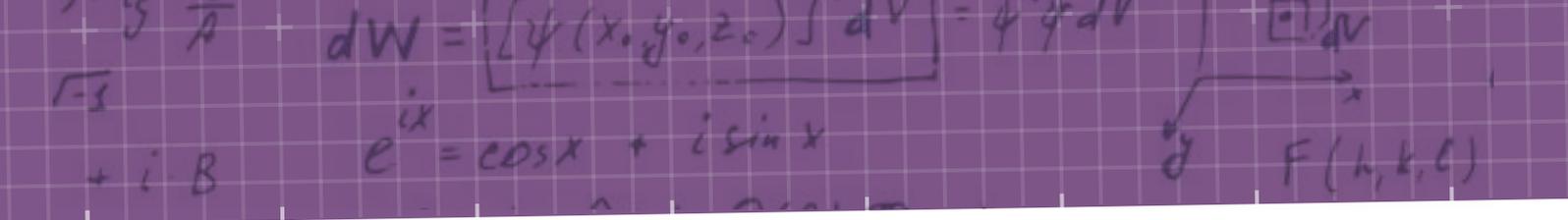


Bild 1. Audioverstärker mit einem LM386N-1 als Prüfling.

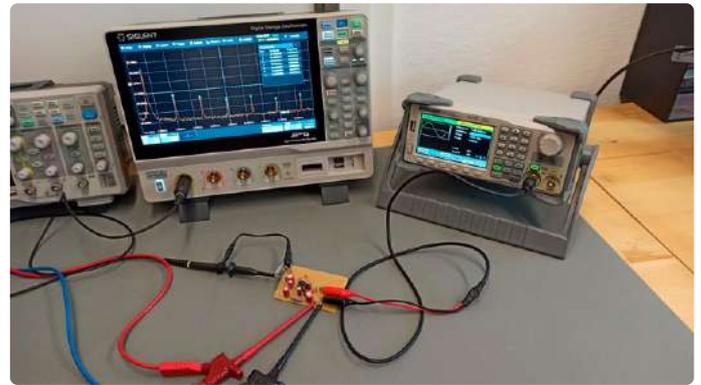


Bild 2. Vollständiger Messaufbau mit Signalgenerator, Oszilloskop und Prüfling.

Der Unterschied zwischen der  $THD_F$  und dem  $THD_R$  ist umso kleiner, je kleiner die Werte selbst sind. Unter 1 % ist der Unterschied nahezu vernachlässigbar.

In Stromversorgungsnetzen wird aus naheliegenden Gründen die Netzfrequenz, hierzulande also 50 Hz, als Grundfrequenz bei der Berechnung der THD verwendet. In der Audiotechnik ist 1 kHz ein üblicher Standard.

In der Praxis werden selbstverständlich niemals auch nur näherungsweise unendlich viele Oberwellen berücksichtigt. Beispielsweise werden maximal 50 Oberwellen in Stromversorgungsnetzen berücksichtigt, in der Audiotechnik ist es sinnvoll, nicht hörbare Oberwellen zu vernachlässigen.

### Praxisbeispiel

Um zu zeigen, dass die graue mathematische Theorie in der Praxis gar nicht so schlimm ist, soll folgendes Beispiel betrachtet werden: Ein Sinussignal mit einer Frequenz von 1 kHz wird an den Eingang eines Verstärkers (LM386N-1) angeschlossen (**Bild 1**). Das Ausgangssignal des Verstärkers wird an einen 10- $\Omega$ -Widerstand als (impedanztechnisch nicht ganz optimalen) Lautsprecherersatz gelegt und via 1:10-Tastkopf an ein HD-Oszilloskop SDS2104X von Siglent geführt (**Bild 2**).

In der Zeitdomäne betrachtet sieht das Ausgangssignal zunächst

ziemlich sauber und sinusförmig aus (**Bild 3**). Dieser erste Eindruck täuscht jedoch, wie ein Blick auf das FFT-Spektrum deutlich zeigt: Die Oberwellenanteile – hier auszugsweise bis 7 kHz – sind neben der Grundwelle bei 1 kHz deutlich sichtbar (**Bild 4**).

Mein Versuch, die Zahl der berücksichtigten Oberwellen zumindest bis zum Bandbreitenlimit des Oszilloskops der Unendlichkeit anzunähern, wurde aus Gründen der Übersichtlichkeit verworfen. Erfasst wurden die Leistungspegel der ersten sieben Harmonischen, entsprechend dem Leistungspegel der Grundwelle und der ersten sechs Oberwellen. Die meisten Oszilloskope haben die Möglichkeit, die gemessenen Leistungspegel automatisch übersichtlich in einer Tabelle anzuzeigen. Dabei erfolgt die Anzeige der Leistungspegel üblicherweise in der Einheit dBm. Hierbei handelt es sich um das zehnfache des dekadischen Logarithmus des Verhältnisses des gemessenen Wertes in Bezug auf eine Referenzgröße. Im Fall von dBm ist die Referenzgröße 1 mW. Bevor gerechnet werden kann, müssen die absoluten Leistungspegel in eine Signalpegeldifferenz relativ zur Grundwelle umgerechnet werden. Hierbei kommt die Einheit dBc (dB carrier, dB relativ zum Träger) zur Anwendung. Zur Umrechnung muss jeweils der Leistungspegel der Grundwelle von den Leistungspegeln aller Harmonischen subtrahiert werden. Beispielsweise für die 2. Harmonische (= 1. Oberwelle):  $-31,7 \text{ dBm} - 22,1 \text{ dBm} = -53,8 \text{ dBc}$ .

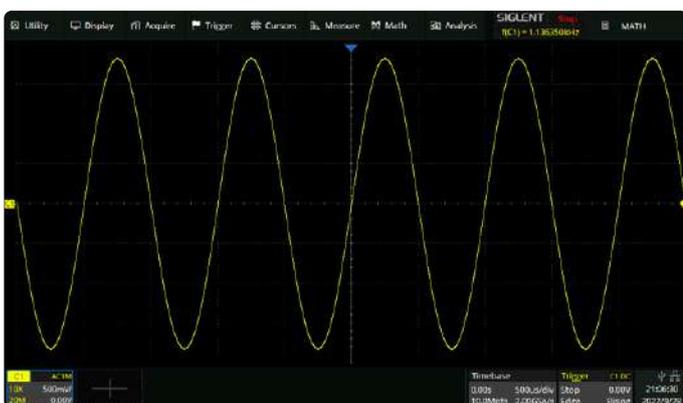


Bild 3. Ausgangssignal des Prüflings in der Zeitdomäne.



Bild 4. FFT-Spektrum des Ausgangssignals.

**Tabelle 1. Absolute und relative Amplituden der ersten sieben Harmonischen (gerundet).**

Ordnungszahl der Harmonischen	Leistungspegel (dBm)	Relative Leistungspegel (dBc)
1. (1 kHz) = Grundwelle	22,1	0
2. (2 kHz) = 1. Oberwelle	-31,7	-53,8
3. (3 kHz) = 2. Oberwelle	-38,7	-60,8
4. (4 kHz) = 3. Oberwelle	-43,8	-65,9
5. (5 kHz) = 4. Oberwelle	-63,5	-85,6
6. (6 kHz) = 5. Oberwelle	-47,0	-69,1
7. (7 kHz) = 6. Oberwelle	-58,0	-80,1

**Tabelle 1** fasst die Harmonischen mit ihren Ordnungszahlen, die absoluten Leistungspegel in dBm (gerundet) und die errechneten Leistungspegeldifferenzen in Bezug auf die 1. Harmonische übersichtlich zusammen. Die relativen Leistungspegel aus Tabelle 1 können direkt in die Formel 1 eingesetzt werden.

$$100 \cdot \sqrt{10^{\frac{-53,8}{10}} + 10^{\frac{-60,8}{10}} + 10^{\frac{-65,9}{10}} + 10^{\frac{-85,6}{10}} + 10^{\frac{-69,1}{10}} + 10^{\frac{-80,1}{10}}} \approx 0,232\%$$

Der THD<sub>F</sub> des Verstärkers (bis 7 kHz betrachtet) beträgt somit circa 0,232 %. Mit der zweiten Formel kann weiter die THD<sub>R</sub> (Klirrfaktor) errechnet werden.

$$THD_R = \frac{0,232}{\sqrt{1 - 0,232^2}} \approx 0,239\%$$

Der Klirrfaktor beträgt folglich etwa 0,239 %. Das Datenblatt des LM386 nennt eine THD von 0,2 %. Dies ist ziemlich nah an den hier errechneten Werten. Texas Instruments verrät im Datenblatt leider nicht, um welchen THD-Wert es sich hierbei handelt, noch wird verraten, wie viele Oberwellen in die Messung eingeflossen sind.

### Grenzen und Hinweise

Die hier vorgestellte Methode findet ihre Grenzen im Dynamikumfang des im jeweiligen Oszilloskop verwendeten Analog-Digital-Umsetzers (ADC). Typisch sind 8-Bit-ADCs mit einem theoretischen Dynamikumfang von rund 48 dB. Diese sind bis zu einem minimalen THD von ungefähr 3% nutzbar.

Dem aufmerksamen Leser wird aufgefallen sein, dass der oben beispielhaft ausgerechnete THD deutlich unter dieser Grenze liegt. Dies ist möglich, weil von mir ein 12-Bit-Oszilloskop verwendet wurde. Mit 12-Bit-Oszilloskopen beträgt der theoretische Dynamikumfang immerhin rund 72 dB. Hiermit sind THD-Werte ab etwa 0,2 % noch im Rahmen des Möglichen.

Um diese Grenzen bestmöglich nutzen zu können, ist jedoch Vorsicht geboten: Der tatsächlich verfügbare Dynamikumfang hängt maßgeblich auch von der eingestellten vertikalen Ablenkung ab! Wird der ADC übersteuert, entstehen Verzerrungen, wird er nicht ausreichend angesteuert, geht wertvoller Dynamikumfang verloren.

Wer diese Methode direkt ausprobieren möchte, fragt sich eventuell, woher er Referenzsignale mit bekannten THD-Werten herbekommt. Hier kann ein einfacher Signalgenerator weiterhelfen: Ein einigermaßen symmetrisches Rechtecksignal mit einem Tastverhältnis von 50 % weist einen THD<sub>F</sub>-Wert von rund 48,3 % auf. Bei einem ebenso symmetrischen Dreiecksignal mit gleichem Tastverhältnis beträgt die THD<sub>F</sub> noch rund 12,1 %.

Wer etwas tiefer in die Materie einsteigen möchte und auch die Herleitung der Formeln verstehen möchte, findet unter [2] und [3] weitere Informationen. 

220398-02

### Sie haben Fragen oder Kommentare?

Sie haben technische Fragen oder Kommentare zu diesem Artikel? Gerne können Sie sich an den Autor des Artikels wenden unter der E-Mail-Adresse [sebastian@baltic-lab.com](mailto:sebastian@baltic-lab.com) oder an die Elektor-Redaktion unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

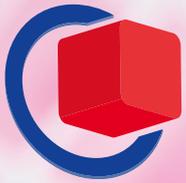


### Passende Produkte

- > **Siglent SDS1204X-E 4-Kanal-Oszilloskop (200 MHz) (SKU 18524)**  
[www.elektor.de/18524](http://www.elektor.de/18524)
- > **Siglent SDG1032X 2-Kanal-Signalgenerator (30 MHz) (SKU 20276)**  
[www.elektor.de/20276](http://www.elektor.de/20276)
- > **Adafruit 2,5 W Class D Mono-Verstärker (PAM8302) (SKU 18745)**  
[www.elektor.de/18745](http://www.elektor.de/18745)

### WEBLINKS

- [1] Doron Shmilovitz, „On the definition of total harmonic distortion and its effect on measurement interpretation“, Power Delivery, IEEE Transactions, S. 526 – 528, 2005: <https://doi.org/10.1109/TPWRD.2004.839744>
- [2] Sebastian Westerhold, „Total Harmonic Distortion (THD) from dBc“, 2022: <https://baltic-lab.com/thd/>
- [3] Sebastian Westerhold, „Total Harmonic Distortion (THD) analysis utilizing the FFT capabilities of modern digital storage oscilloscope“, 2022: <http://dx.doi.org/10.5281/zenodo.6969825>



# embeddedworld2023

Exhibition&Conference

... it's a smarter world

JOIN THE EMBEDDED  
COMMUNITY

14.-16.3.2023



Get your  
free ticket now!

[embedded-world.de/gutschein](https://embedded-world.de/gutschein)

Use the voucher code **GG4ew23**

Medienpartner

Markt&Technik  
DES UNABHÄNGIGE WIRTSCHAFTSZEITUNG FÜR ELEKTRONIK

Elektronik

SmarterWorld  
Solutions for a Smarter World

DESIGN &  
ELEKTRONIK  
KNOW-HOW FÜR ENTWICKLER

Elektronik  
automotive

•medical-design

computer &  
automation

elektroniknet.de

NÜRNBERG MESSE



# Allsehende Maschinen

Die Technologie hinter modernen industriellen Bildverarbeitungssystemen

Von Stuart Cording (Elektor)

Mit dem Aufkommen von CMOS-Bildsensoren können Low-Power-Kameras jetzt problemlos in einem wirtschaftlichen Halbleiterherstellungsprozess in Massenproduktion hergestellt werden. Heute tragen wir alle mindestens zwei solcher Geräte in unseren Smartphones. Dank immer leistungsfähigerer Mikrocontroller stehen Bilderfassung und Objektverarbeitung allen zur Verfügung und unterstützen sogar 3D-Messungen in Industriesensoren.

Kameras sind in der heutigen Gesellschaft immer in der Nähe. Wenn Sie keine in der Tasche haben, ist es wahrscheinlich, dass die Person, neben der Sie im Zug sitzen, eine hat. Und wenn nicht, gibt es wahrscheinlich eine im Waggon, im Bahnhof oder im Sicherheitssystem des Cafés, in dem Sie auf den Zug gewartet haben. Kameras sind dank CMOS-Bildsensoren so allgegenwärtig geworden, weil sie im Vergleich zur alten CCD-Technologie (Charge-Coupled Devices) preiswert sind, kleiner sind und mit weniger Energie auskommen. Während das Aufnehmen von Bildern und Videos eine traditionelle Aufgabe ist, hat das schnelle Wachstum der Rechenleistung im letzten Jahrzehnt es Computern ermöglicht, einlaufende visuelle Daten in Echtzeit zu analysieren. Dies ermöglicht eine automatisierte Inspektion für verschiedene industrielle Anwendungen, zum Beispiel das Erkennen falsch platzierter Bauteile auf Leiterplatten, das Lokalisieren von Teilen auf Förderbän-

dern und sogar das Beurteilen der Qualität einer Schweißverbindung.

## CCDs - Bildtechnologie der Vergangenheit

Bildsensoren verwenden ein Array von Fotodioden, um Photonen in elektrischen Strom umzuwandeln. Bei CCDs wird eine Reihe von Fotodioden über Schalter verbunden. Nachdem Licht auf die Sensorfläche aufgetroffen ist, werden die in jedem Pixel vorhandenen Ladungen durch die Kette (Ladungsdurchgang) eine nach der anderen zum Rand des Sensors weitergeleitet (**Bild 1**). Hier wird die Ladung jeder Diode in eine Spannung umgewandelt, bevor sie ein Analog-Digital-Wandler (ADC) in einen digitalen Wert umrechnet. CCD-Sensoren werden in einem Metall-Oxid-Halbleiter-Prozess (MOS) hergestellt, der sich von der für die meisten Silizium-Bauelemente verwendeten komplementären MOS-Technologie (CMOS) unterscheidet.

Dass CCD als Sensortechnologie überhaupt schon seit langer Zeit bis heute eingesetzt wird, hängt mit ihrer hohen Quanteneffizienz zusammen, einem Maß dafür, wie gut das Bauteil Photonen in Elektronen umwandelt, und die bis zu 95 % betragen kann. Neben kommerziellen und Consumer-Kameraanwendungen wurden CCDs in wissenschaftlichen Geräten wie Spektrometern und sogar dem Hubble-Teleskop verwendet [1]. Die Auswertung der aufgenommenen Bilder erfordert jedoch externe Schaltungen.

CMOS-Bildsensoren entwickelten sich erst in den 1990er Jahren zu einer praktikablen Technologie. Anfangs war die Bildqualität schlecht, mit niedriger Auflösung, verrauschten Bildern und schlechter Farbdefinition. Aufgrund notwendiger Konstruktionsunterschiede waren die Photodioden dieser Sensoren wesentlich kleiner als die eines CCD-Sensors mit gleicher Pixelzahl. Anstatt einer Ladungsschieber-Schaltung benötigte jedes Pixel einen eigenen Verstärker, wodurch die für jedes Pixel verfügbare Oberfläche reduziert wurde (**Bild 2**). Dank der immensen Entwicklung und Forschung, die klar aufgrund von Marktbedürfnissen betrieben wurde, sind die Unterschiede in der Bildqualität zwischen den beiden Technologien heute jedoch marginal. Die einfache Produktion, die sich aus der Nutzung eines CMOS-Fertigungsprozesses ergibt, hat auch die Preise gedrückt und die Integration eines Großteils der erforderlichen Bildverarbeitungsunterstützung auf dem Sensorchip verbessert. Darüber hinaus ist der Strombedarf etwa 100-mal niedriger als

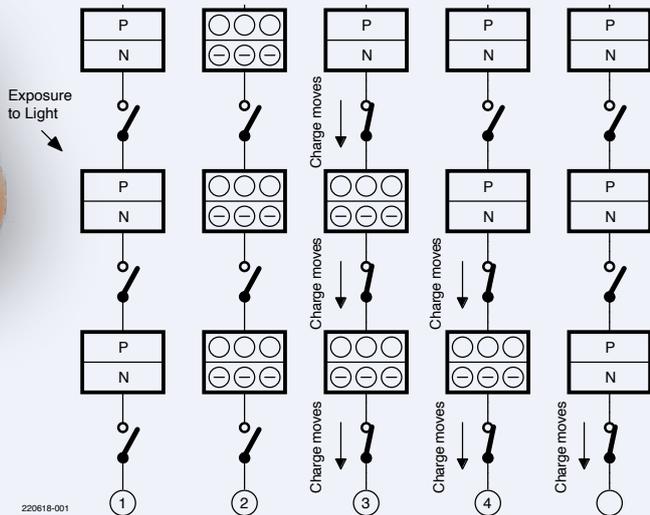


Bild 1. CCD-Bildsensoren nutzen die Ladungsübertragung zwischen Pixeln, um ihre Werte auszulesen. (Quelle: Canon Inc.)

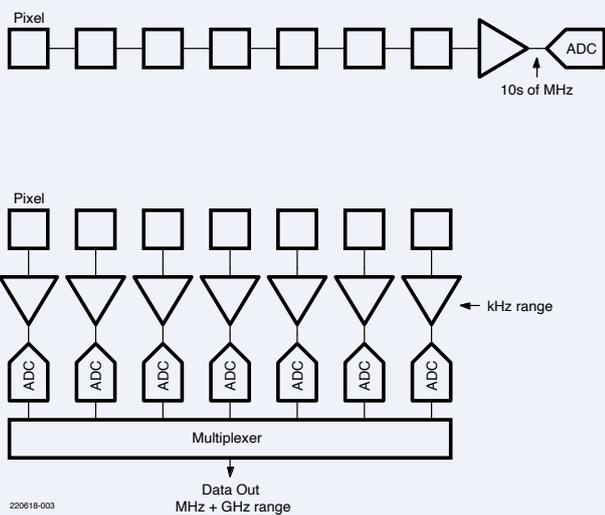


Bild 3. Da CMOS-Bildsensoren massiv parallel sind, muss das analoge Frontend nicht so schnell sein wie bei CCD-Sensoren. (Quelle: Teledyne Digital Imaging Inc.)

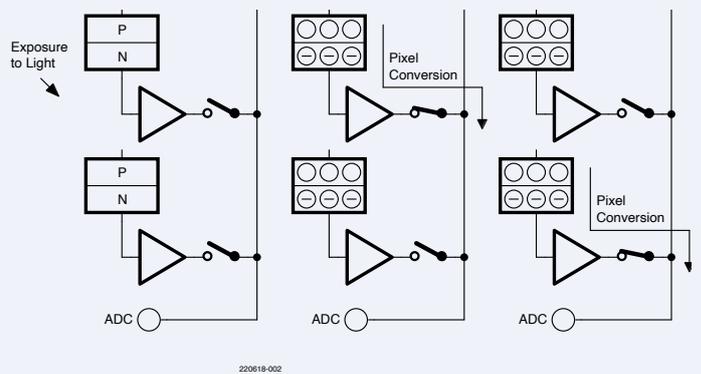


Bild 2. CMOS-Bildsensoren verwenden einen Analog/Digital-Wandler, um den Wert jedes Pixels einzeln umzuwandeln. (Quelle: Canon Inc.)

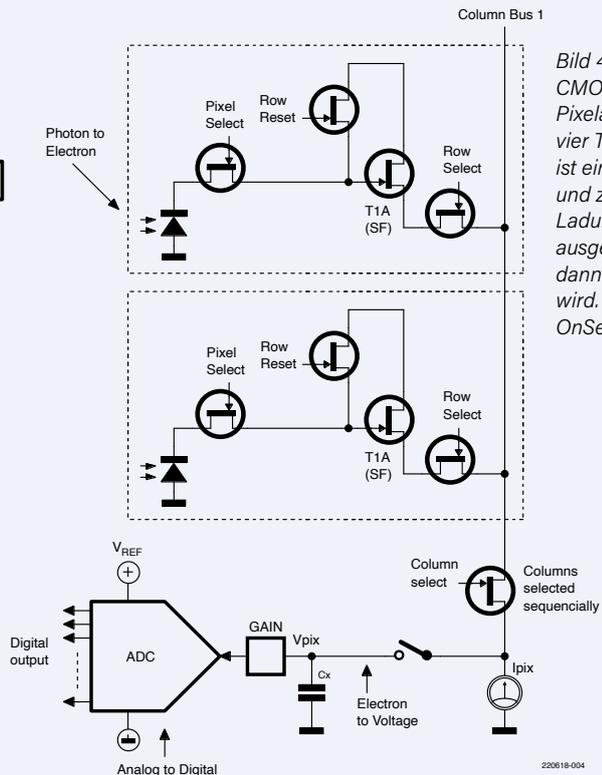


Bild 4. Diese CMOS-Pixelarchitektur mit vier Transistoren ist eine von vielen und zeigt, wie die Ladung des Pixels ausgewertet und dann zurückgesetzt wird. (Quelle: OnSemi)

bei einem vergleichbaren CCD-Sensor, was für moderne batteriebetriebene mobile Geräte von entscheidender Bedeutung ist.

Die Reduzierung des Pixelrauschens und die Erhöhung der Frame-Rate werden durch eine massiv parallele Datenkonvertierung erreicht, wobei mehrere ADCs die Ladungen in einer Pixelzeile gleichzeitig konvertieren. Im Vergleich zu CCDs ermöglicht dies die Verwendung von Verstärkern und ADCs mit geringerer Bandbreite (**Bild 3**). Die Pixeldesigns variieren, bestehen jedoch normalerweise stets aus einem hochohmigen Verstärker, der mit dem Fotodioden-Übergang verbunden ist, und Schaltern für die Zeilen- und Spaltenauswahl. Schließlich ist ein Schalter vorgesehen, um die Ladung auf dem Pixel zu löschen (**Bild 4**).

## Farbe hinzufügen

Zwar mag der beliebte CMOS-Produktionsprozess die Kosten für Bildsensoren gesenkt haben mag, alles andere ist aber bei CMOS-Sensoren alles andere als normal. Die Fotodioden sind für sämtliches sichtbare Licht empfindlich, sodass jedes aufgenommene Bild nur ein Graustufenbild liefern würde. Um Farbbilder zu erzeugen, wird ein Farbfilerarray (CFA) benötigt. Dazu wird üblicherweise ein Bayer-Filter genutzt, benannt nach Bryce Bayer von Eastman Kodak, der es erfunden hat. Es verwendet rote, blaue und zwei grüne Farbfiler in einem Muster so, dass jedes Quadrat aus vier Pixeln halb grün, ein Viertel rot und ein Viertel blau ist. Die höhere Gewichtung von Grün ist auf die Physiologie

des menschlichen Auges zurückzuführen, das Grün besser wahrnimmt [2].

Das resultierende Rohbild ist ohne Vorverarbeitung nicht für den unmittelbaren menschlichen Konsum geeignet. Dies wird erst durch einen Prozess erreicht, der als Entmosaikierung bezeichnet wird. Im einfachsten Fall wird jedes Pixel basierend auf den Werten bestimmter umgebender Pixel in einen RGB-Wert (Rot/Grün/Blau) umgewandelt. Allerdings können einige Entmosaikierungsansätze wie lineare Interpolation Artefakte im endgültigen Bild verursachen, insbesondere an den Rändern von Objekten. Um dem entgegenzuwirken, steht eine Auswahl an Algorithmen zur Verfügung, die Artefakte minimieren und gleichzeitig ein schnelles endgültiges Bild liefern sollen [3].

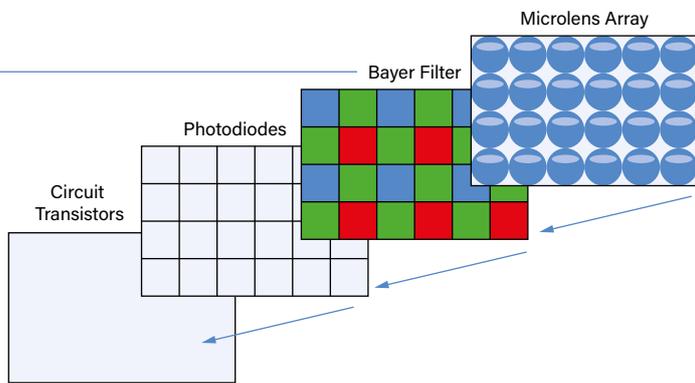


Bild 5. Im Gegensatz zu typischen Halbleitern benötigen Bildsensoren ein Farbfilter und ein Mikrolinsen-Array.

Ein weiterer Unterschied zur typischen Herstellung von „richtigen“ Chips ist die Mikrolinsenschicht. Da der lichtempfindliche Bereich jedes Pixels so klein ist, hilft die Linse, das einfallende Licht auf das Pixel zu fokussieren, was wiederum die Quanteneffizienz erhöht (Bild 5).

### Elektronische Verschlüsse

Auch die Funktionsweise des Bildsensor-Verschlusses (Shutter) kann für Laien schwer verständlich sein. Ein elektronischer Verschluss wird verwendet, um die Pixel dem Licht für die richtige Zeitspanne auszusetzen. Typischerweise sind zwei Hauptoptionen verfügbar, global und rollend, und einige Bildsensoren unterstützen beide Verschlussarten. Bei einem *Global Shutter* werden alle Pixel des Bildsensors elektronisch so gesteuert, dass sie gleichzeitig belichtet werden. Anschließend wird jede Pixelzeile in digitale Werte umgewandelt und an den Host-Prozessor weitergeleitet, der die Bildverarbeitung durchführt. Beim *Rolling Shutter* dagegen wird jede Pixelreihe nacheinander belichtet, wobei die Konvertierung für jede Zeile unmittelbar erfolgt.

Der sich ergebende Unterschied im Bild wird teilweise durch die verwendete Verschlussart beeinflusst. Ein Global Shutter fängt das Bild in einem einzigen Moment ein. Im Vergleich dazu erfasst der Rolling Shutter jede Zeile mit einer leichten Verzögerung gegenüber der vorherigen Zeile. Wenn sich das vom Sensor erfasste Objekt (schnell) bewegt, können Artefakte im resultierenden Bild auftreten. Wenn Sie also beispielsweise eine Lokomotive fotografieren, die sich von links nach rechts bewegt, zeigt das endgültige Bild eine Bewegungsunschärfe von links nach rechts, beginnend am oberen Rand des Bildes. In einem solchen Fall scheint es offensichtlich besser, einen globalen Verschluss auszuwählen, doch wie so oft steckt der Teufel im

Detail. Der Rolling-Shutter-Ansatz erfordert nämlich weniger Transistoren um jedes Pixel herum, wodurch mehr Fläche für den lichtempfindlichen Teil übrig bleibt, was wiederum zu einer verbesserten Bildqualität und zu geringerem Pixelrauschen führt. Außerdem erreicht der Rolling-Shutter höhere Frame-Raten als ein Global-Shutter-Design, da die oberen Zeilen des nächsten Frames mit der Belichtung beginnen können, während die letzten Zeilen noch umgewandelt werden. Die Entscheidung, was besser ist, hängt also von der konkreten Anwendung ab. Die Probleme, die durch den Rolling Shutter verursacht werden, können durch einen mit dem Shutter synchronisierten Blitz gelöst werden. Einige wissenschaftliche Instrumente erfordern einen Global-Shutter-Ansatz, während eine visuelle Inspektionskamera mit einem Rolling-Shutter und einem synchronisierten Blitzlicht zufrieden ist. Eine hilfreiche Ressource, um die Probleme genauer zu verstehen, wird von der PCO AG [4] bereitgestellt.

Eine weitere potenzielle Kuriosität bei Bildsensoren ist der Unterschied zwischen der Anzahl der Pixel auf dem Sensor und der Anzahl, die für die Anwendung des Benutzers verfügbar ist. Zum Beispiel verfügt der OnSemi AR0130CS [5], ein digitaler 1/3-Zoll-CMOS-Bildsensor mit Rolling Shutter und 1,2 Megapixel laut Datenblatt über 1280 × 960 aktive Pixel. Der Sensor hat jedoch tatsächlich 1412 × 1028 Pixel mit einer aktiven Fläche von 1288 × 968. Einige der oberen Zeilen und ein Spaltenblock auf der rechten Seite sind optisch schwarz. Diese Bereiche werden intern zur Überwachung des Schwarzpegels und zur Schwarzpegelkorrektur verwendet (Bild 6).

Eine weitere Herausforderung, eine Aufnahme in guter Bildqualität zu erreichen, liegt in der Stromversorgung. Das Stromversorgungsunterdrückungsverhältnis (power supply rejection ratio, PSRR) der verwendeten Low-Dropout-Regler (LDO) muss so hoch

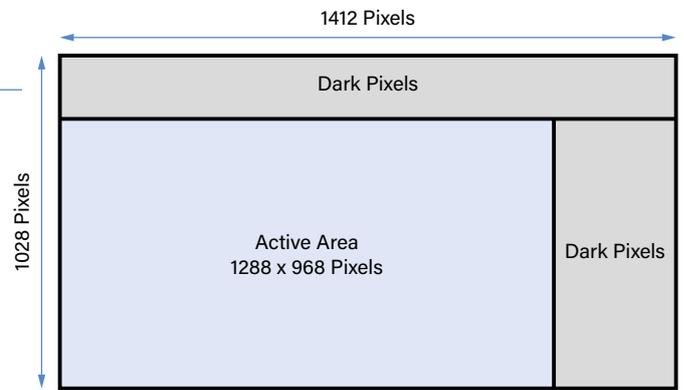


Bild 6. In einem CMOS-Bildsensor können zusätzliche Pixel implementiert werden, um eine Schwarzpegelkompensation bereitzustellen. (Quelle: OnSemi)

sein, dass jegliches Rauschen von den Versorgungsanschlüssen des CMOS-Bildsensors ferngehalten wird. Wenn nicht, ist das Risiko von Pixelrauschen erhöht. Da Bildsensoren über 50 Megapixel hinausgehen, wird laut einer Application Note von OnSemi [6] ein PSRR von größer 90 dB bis 10 kHz und größer 45 dB zwischen 1 MHz und 3 MHz empfohlen. Auch die dynamische Last kann mit einigen hundert Milliampere Laständerung bei den Übergängen der Reihen recht beträchtlich sein. Hier sind Bulk-Kondensatoren unerlässlich, um eine optimale Leistung zu erzielen. Schließlich sollte das Ausgangsrauschen des LDO (nV/√Hz) berücksichtigt werden, da alles im Bereich von 10 Hz bis 1 MHz zu zusätzlichem Rauschen in den Pixeln führen und deren Dynamikbereich verringern kann.

### Machine Vision mit Mikrocontrollern

Mit ihren 32-Bit-Prozessoren und einem Megabyte SRAM oder mehr eignen sich die größeren Mikrocontroller von heute gut für einfache Bildverarbeitungsanwendungen und können sogar einige nützliche Algorithmen für maschinelles Lernen (ML) zur Objekterkennung enthalten. Ein in Betracht zu ziehendes Projekt ist OpenMV [7], eine Kombination aus Hardware- und Software-Entwicklungsumgebung, die speziell auf maschinelles Sehen abzielt. Mit der Mikrocontroller-Familie STM32 bietet die Hardware Zugriff auf Bildsensoren mit bis zu 5 Megapixeln.

Die Softwareentwicklung wird durch das *OpenMV Integrated Development Environment* unterstützt, das auf die Bedürfnisse der Bildverarbeitung abgestimmt ist (Bild 7). Auf der linken Seite befindet sich ein Editor zum Schreiben von Code in MicroPython und ein serielles Terminal, das die Textausgabe des Anwendungscodes anzeigt. Auf der rechten Seite dreht es sich um das Kamerabild. Oben befindet sich der Frame-Puffer,

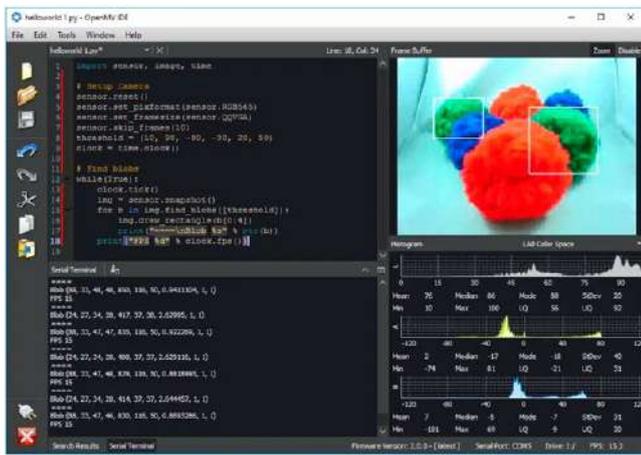


Bild 7. Die OpenMV-Entwicklungsumgebung stellt einen Editor zum Codieren bereit und zeigt das vom angeschlossenen Sensor aufgenommene Bild. (Quelle: OpenMV)

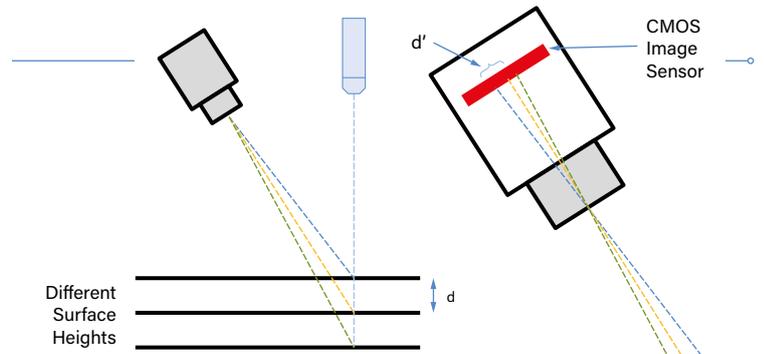


Bild 8. Bei der Lasertriangulation wird ein CMOS-Bildsensor verwendet, um die Reflexion eines Laserstrahls auf einer Oberfläche zur Höhenbestimmung zu erfassen. (Quelle: MoviMED)

der das Bild von der Kamera in Kombination mit allen Ausgaben zeigt, die von der Anwendung in den Videostream eingefügt werden. Dies können Textnachrichten oder Begrenzungsrahmen sein, die mithilfe von ML erkannte Objekte hervorheben. Unten werden Histogramme angezeigt, zum Beispiel die Rot-, Blau- und Grünwerte des Bildes. Ähnlich wie in der Arduino-Umgebung sind Beispielprojekte verfügbar. Dazu gehören auch ML-Modelle für Anwendungen wie Gesichts- und Personenerkennung.

### 3D-Imaging mit Lasertriangulation

Während Kameras mit CMOS-Bildsensoren für eine Vielzahl von Vision-Anwendungen eingesetzt werden können, liegen diese typischerweise im Bereich der 2D-Bildgebung. Eine Vielzahl von Anwendungen erfordert jedoch genaue Oberflächenmessungen, die eine 3D-Metrologiefähigkeit erfordern. Im industriellen Bereich ist die Lasertriangulation eine gängige Lösung, um die Oberfläche von fertigen Produkten wie Schweißnähte, Klebrauhen und die Position von Bauteilen auf Platinen zu vermessen.

Der Ansatz für diese Sensoren besteht darin, einen Laserpunkt oder eine Laserlinie auf das Messobjekt zu projizieren. Von der Oberfläche reflektiertes Licht trifft auf einen integrierten CMOS-Sensor, der mittels Triangulation die Entfernung zum Objekt bestimmt (Bild 8). Wie der Bildsensor und der Laser positioniert sind, bestimmt die Komplexität der Berechnungen, die zum Bestimmen des Abstands, der Auflösung und der Wahrscheinlichkeit von Messfehlern erforderlich sind. Beispielsweise kann der Laser senkrecht zum Objekt (Standardgeometrie) mit der Kamera in einem Winkel sein, was einen guten Allzwecksensor ergibt. Beim Messen von stark reflektierenden Oberflächen wie Glas und poliertem Metall ist jedoch eine Look-Away-Geometrie

sinnvoll, bei der sowohl der Bildsensor als auch der Laser in einem Winkel im Vergleich zur Messstelle angebracht sind. Sensorhersteller wie SmartRay bieten auf Basis dieser Technologie ein breites Spektrum an 3D-Sensoren an. Ihre Sensorreihe ECCO 95.015G [8] wurde speziell entwickelt, um Flachglas und spiegelnde, stark reflektierende Oberflächen zu scannen. In einem Abstand von 23,5 mm zum Messobjekt montiert, bieten sie eine vertikale Auflösung von 0,42...0,54 µm. Das Gerät liefert Messwerte als Reihe von Punkten über eine Gigabit-Ethernet-Schnittstelle, die mit typischer industrieller Bildverarbeitungssoftware zur Auswertung der Daten verbunden werden kann.

### CMOS-Bildsensoren für alle

Aufgrund ihrer geringen Kosten und ihres geringen Stromverbrauchs sind CMOS-Bildsensoren in der modernen Gesellschaft allgegenwärtig geworden. Sensoranwendungen, bei denen Ultraschall- oder Infrarotsensoren eingesetzt worden wären, könnten heutzutage durchaus CMOS-Bildsensor verwendet werden. Bei Millionen von Pixeln

und der enormen und schwer verständlichen Funktionalität kann es jedoch schwierig sein, sie mit Mikrocontrollern zu verbinden. Glücklicherweise sind Rechenleistung und Speicherkapazität heutiger Mikrocontroller in der Lage, relativ komplexe maschinell lernende Objekterkennung in Verbindung mit Bildsensoren von bescheidener Größe zu handhaben. Und bei diesen Geräten geht es nicht nur um 2D-Bilder. Industrielle Sensoren mit laserbasierter Triangulation verfügen über CMOS-Bildgebungsbausteine, um mikrometeregenaue Messungen in drei Dimensionen zu liefern und die visuelle Inspektion von Werkstücken zur Qualitätssicherung zu automatisieren. ◀

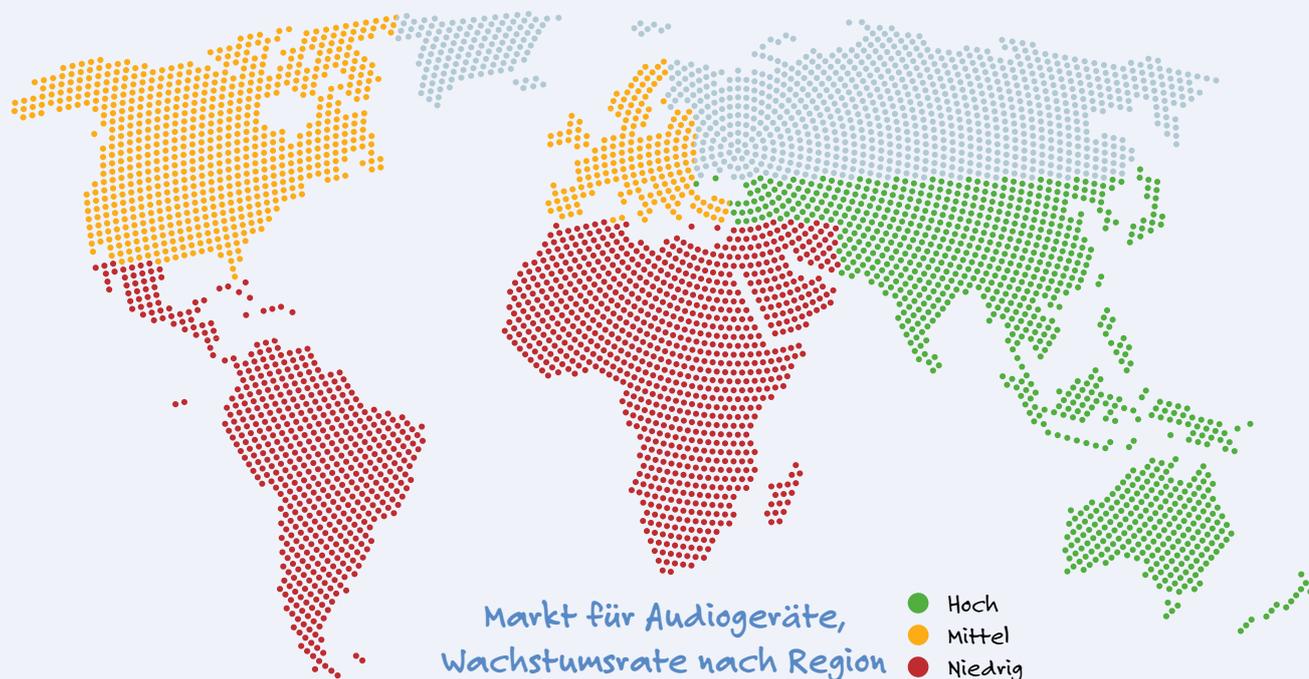
(220618-02)RG

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Senden Sie eine E-Mail an den Autor unter [stuart.cording@elektor.com](mailto:stuart.cording@elektor.com) oder kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

### WEBLINKS

- [1] „Hubble’s Instruments: WFC2 Wide Field Planetary Camera 2“, European Space Agency: <https://bit.ly/3WBhN1Z>
- [2] „The Human Eye’s Response to Light“, Iowa State University: <https://bit.ly/3UsmVDU>
- [3] A. Rajwade, „Color Image Demosaicing“, Indian Institute of Technology Bombay: <https://bit.ly/3UneoCg>
- [4] „What are all the discussions about global vs. rolling shutter?“, PCO AG: <https://bit.ly/3h90h4R>
- [5] Produktseite CMOS-Bildsensor AR0130CS: <https://bit.ly/3hgWq63>
- [6] M. Dadafshar, „Understanding Challenges in Powering High Resolution, High Frame Rate CMOS Image Sensors“, OnSemi, Juli 2022: <https://bit.ly/3Dz7Ets>
- [7] Website OpenMV: <https://openmv.io/>
- [8] Produktseite ECCO 95.015G: <https://bit.ly/3E7HakZ>



## Der asiatisch-pazifische Raum übernimmt die Führung

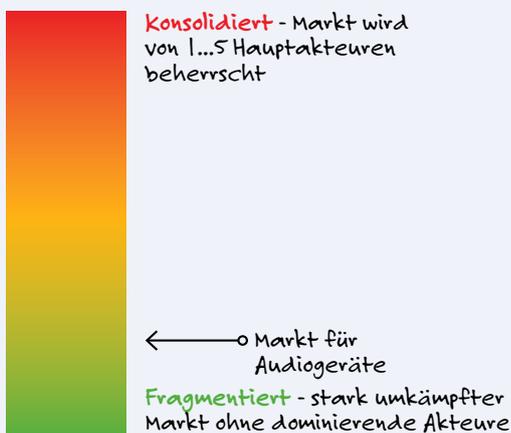
Für Audiogeräte im professionellen Umfeld - Mischpulte, Verstärker, Mikrofone und Lautsprecher/Monitore - rechnen Experten in den nächsten vier Jahren mit einer Wachstumsrate von etwa 6 %. Dies gilt auch für alle Audiogeräte im Automobilsektor. Der Markt wird von rund 13,5 Mrd. US-Dollar im Jahr 2022 auf

18 Mrd. US-Dollar im Jahr 2027 wachsen. Der asiatisch-pazifische Raum ist die am schnellsten wachsende Region, gefolgt von den Regionen Europa und Nordamerika. Der Unterschied in den Wachstumsraten zwischen dem asiatisch-pazifischen Raum, Europa/Nordamerika und Afrika/Südamerika lässt sich zu

einem großen Teil durch den boomenden Automobilmarkt in China und Indien erklären. Ein einziges Auto kann mit bis zu zehn Lautsprechern ausgestattet sein – oder sogar mehr.

(Quellen: Fortune Business Insights, Mordor Intelligence - map)

### Marktkonzentration Audio-Equipment



Quelle: Mordor Intelligence (Bericht Audio Equipment Market, 2022-2027)

## Marktkonzentration? Nicht so sehr!

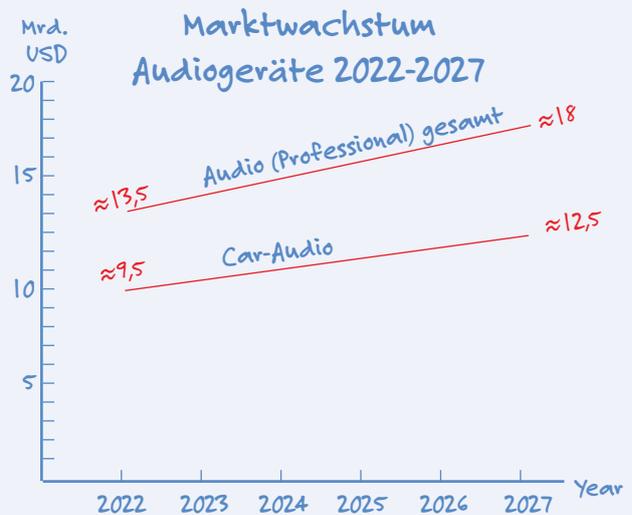
Der Markt für professionelle Audiogeräte hat eine glänzende Zukunft. Folglich mangelt es den Anbietern nicht an Interesse, diesen Markt zu erkunden. Dies wiederum verbessert die Aussichten für die Käufer von Audiogeräten noch mehr. Die relativ geringe Marktkonzentration bedeutet, dass der Preiswettbewerb auf dem Markt für Audiogeräte sicherlich hart sein wird. Außerdem gibt es viele Innovationen, sei es im Bereich der Verstärker, Mikrofone oder Lautsprecher. Kurz gesagt: Der professionelle Kunde kann eine qualitativ hochwertige Tonaufnahme und -ausgabe zu vernünftigen Preisen erwarten. Die Audio-Innovation wird auch den Videokonferenzen Auftrieb geben.

(Quellen: Microsoft, Mordor Intelligence)

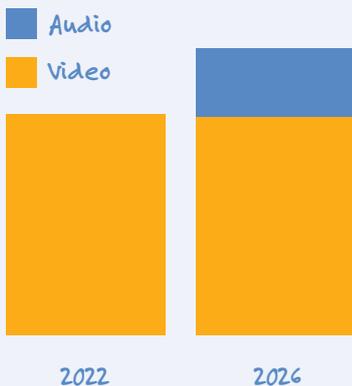
# Das Auto der Zukunft ist... Sound Out/Sound In

Wenn man einige Zahlen zusammenzählt, wird deutlich, dass der Markt für professionelle Audiogeräte hauptsächlich Audiogeräte in Autos betrifft. Sehen Sie sich bitte die Grafik an. Warum ist das Auto hier so vorherrschend? Der erste Grund ist, dass Autofahrer einen Großteil ihrer Zeit im Auto (im Stau) verbringen. Schätzungen gehen von etwa 45 Stunden pro Jahr aufgrund von Verkehrsbehinderungen aus. Der zweite Grund ist, dass die Nutzung von Infotainment-Diensten in einem Auto gefährlich sein kann: Sie können den Fahrer davon ablenken, sich auf das Geschehen auf der Straße zu konzentrieren. Aus diesem Grund ist die Spracherkennung so wichtig. Für die Spracherkennung werden Audioprozessoren benötigt, und diese sind ebenfalls Teil des Marktes für Audiogeräte.

(Quellen: Fortune Business Insights, Mordor Intelligence)



## Ein Überblick: Video schlägt Audio



Es gibt einen Unterschied zwischen dem professionellen und dem gesamten Audiomarkt. Der letztere umfasst nämlich die Heimanwender. Der gesamte Audiomarkt umfasst alle Arten von Rekordern und Wiedergabegeräten, nicht nur professionelle Mischpulte und ähnliches. Das Gleiche gilt für Videogeräte: Eine Fernsehkamera ist etwas anderes als eine Smartphone-Kamera. Wenn wir jedoch die professionelle Nutzung und den Hausgebrauch sowohl für Audio als

auch für Video zusammenfassen, wie sehen dann die Zahlen aus? Nun, dieser Markt war im vergangenen Jahr etwa 158 Milliarden US-Dollar wert, mit einem prognostizierten Wachstum auf 196 Milliarden US-Dollar im Jahr 2026. Etwa 80 % des Gesamtmarktes entfallen auf Videogeräte. Curved TVs werden in den kommenden Jahren zu den Gewinnern gehören.

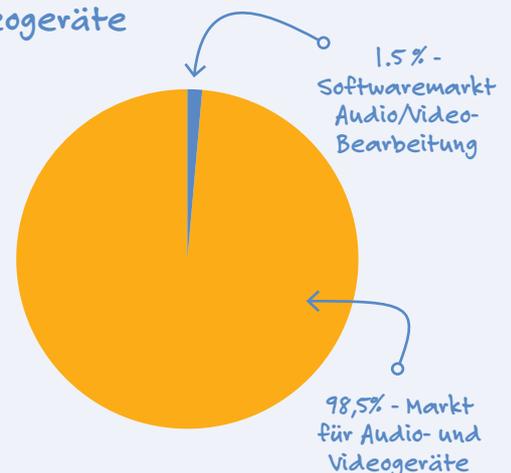
(Quellen: Market Prospects, Research and Markets)

## Hardware? OK, aber was ist mit der Software?

Bis jetzt haben wir nur über Hardware gesprochen. Was ist mit Software? Natürlich gibt es einen Markt für Audio- und Videobearbeitungssoftware, schon allein wegen der YouTuber. Der Softwaremarkt wächst sogar noch schneller als der Markt für Audio- und Videogeräte: 7 % statt rund 6 % in den kommenden Jahren. Im Vergleich zu Audio- und Video-Hardware ist der Markt für Audio- und Video-Software jedoch nur ein Tropfen auf den heißen Stein, denn nach Angaben von Transparency Market Research wird der Markt für Audio- und Videobearbeitungssoftware von rund 2 Mrd. US-Dollar im Jahr 2020 auf etwa 3 Mrd. US-Dollar im Jahr 2030 wachsen. Das bedeutet, dass der Markt für Audio-/Videosoftware nur etwa 1,5 % des Hardwaremarktes ausmacht.

(Quellen: Research and Markets, Transparency Market Research)

### Softwaremarkt Audio/Video-Bearbeitung vs. Markt für Audio- und Videogeräte





# Die Entwicklung der Sprach- und Audiosteuerung für elektronische Geräte



Bild 1. Sprachgesteuerter  
Roboterassistent.  
(Quelle: PaO\_STUDIO  
auf Shutterstock)

Von Mark Patrick (Mouser Electronics)

Sprache ist ein effizientes Mittel für Menschen, um Ideen und Wünsche auszudrücken. Schon vor dem Industriezeitalter fanden die Menschen heraus, dass Tiere darauf trainiert werden können, einfache Befehle zu erkennen und darauf zu reagieren, um eine Aufgabe zu erfüllen. Der nächste logische Schritt bestand darin, Möglichkeiten zu entwickeln, um mit unseren Maschinen zu kommunizieren und ihnen Befehle mit unserer Stimme zu erteilen. Die Verwendung von Sprache und Audiosignalen als Steuerungsschnittstelle für elektronische Geräte ist in den letzten Jahren immer beliebter geworden und entwickelt sich ständig weiter. Damit werden die Erwartungen der Benutzer und die Anforderungen neuer Anwendungen erfüllt. In diesem Artikel erläutern wir die Vorteile der Sprach- und Audiosteuerung von elektronischen Geräten und Maschinen und zeigen, wie sie umgesetzt wird. Wir zeigen auch, wie diese Steuerungsschnittstelle jetzt in Offline-Geräte eingebettet und wie das Audioerlebnis, das sie bieten, erheblich verbessert werden kann.

## Elektronische Geräte per Sprache steuern

Die Sprachsteuerung von Maschinen hat mehrere offensichtliche Vorteile:

- Für Menschen ist Sprache eine intuitive Form der Kommunikation, mit der sie Befehle sehr einfach verbal übermitteln können.
- Die Kommunikation per Sprache ist auch dann möglich, wenn Augen und Hände anderweitig beschäftigt sind. Eine Sprachsteuerung ist also komfortabel und man kann währenddessen auch noch etwas anderes tun. Im Gegensatz dazu ist es unter bestimmten Umständen wie beim Autofahren sogar verboten, gleichzeitig noch ein anderes Gerät zum Beispiel durch Berührung zu bedienen.
- Sprache ist ein effizientes Mittel zur Steuerung von Maschinen, die zuhören und reagieren können, ohne dass komplexe Befehle erforderlich sind.
- Durch die Integration von Sprache wird der Bedarf für einen Touchscreen bei vielen Geräten minimiert. Dies ist vor allem bei entfernten oder tragbaren batteriebetriebenen Geräten von Vorteil, bei denen die Reduzierung der Größe und des Stromverbrauchs häufig eine Herausforderung für das Design darstellen. Der Verzicht auf einen Touchscreen ist auch hygienischer bei Geräten, die von mehreren Benutzern verwendet werden.
- Wie in **Bild 1** dargestellt, kann die Sprachsteuerung ein Hilfsmittel für Menschen mit Behinderung sein, für die eine Touch-Bedienung keine praktikable Option ist. Sprachkommunikation mit Maschinen kann verwendet werden, um beispielsweise Türen zu öffnen oder um aus der Ferne Informationen über den Gesundheitszustand einer Person zu übermitteln.

Das Audio-Front-End (AFE) eines sprachgesteuerten Geräts besteht aus einem Mikrofonsystem und signalverarbeitenden Blöcken. Das AFE verarbeitet das Signal eines Mehrkanal-Mikrofonarrays, um Störungen durch Hintergrundgeräusche oder die vom Gerät selbst emittierten Signale zu eliminieren. Dieses Signal wird dann an eine „Wake Word“-Erkennungs-Engine gesendet, die Wörter wie „Alexa“ oder „OK Google“ erkennt, die auf dem Gerät vorprogrammiert sind. Zur Unterdrückung unerwünschter Störsignale werden mehrere Signalverarbeitungsalgorithmen eingesetzt. Zur Sprachsteuerung zählen folgende Komponenten:

**Mikrofon-Array:** Sprachsteuerungssysteme brauchen ein oder mehrere Mikrofone, um das Audio-Steuersig-

nal zu erfassen. Größe, Kosten, Leistung und Robustheit sind die wichtigsten Faktoren bei der Auswahl eines Mikrofon-Arrays. Die Kombination verschiedener Signale von einem Array mit mehreren Mikrofonen verbessert das Signal-Rausch-Verhältnis (Signal Noise Ratio, SNR) der Audiosignalkette.

**Detektor für die Sprechrichtung (Direction-of-arrival, DoA):** Damit wird die Position des Benutzers in Bezug auf das zu steuernde Gerät bestimmt, damit das Mikrofonarray die Stimme gezielt empfangen kann.

**Beamformer:** Der Beamformer nimmt Geräusche aus der Sprechrichtung auf, während Geräusche aus anderen Richtungen unterdrückt werden. Seine Leistung hängt von der Geometrie und dem Signal-Rausch-Verhältnis des Mikrofonarrays sowie von der Breite des Mikrofonstrahls und dem Pegel der Hintergrundgeräusche ab.

**Akustische Echounterdrückung (Acoustic echo canceller, AEC):** Diese Funktion unterdrückt das Wiedergabesignal auf dem Gerätelautsprecher selbst (etwa wenn der Gerätelautsprecher Musik abspielt), damit der Sprachbefehl des Benutzers klar aufgenommen werden kann.

**Adaptive Interferenzunterdrückung (Adaptive interference canceller, AIC):** Diese Funktion unterdrückt externe Geräusche von anderen Schallquellen, die mit einem herkömmlichen Beamformer nur schwer zu unterdrücken sind, beispielsweise laute Geräusche, die von anderen Geräten erzeugt werden.

**Wake-Word-Detektor:** Das verarbeitete Sprachsignal des Audio-Front-Ends (AFE) wird mit einer Bibliothek von so genannten Wake Words (zum Beispiel „Hey Google“) verglichen, wobei ein Algorithmus zur Erkennung von Wake Words zum Einsatz kommt, der Teil eines Machine-Learning-Modells ist. Größere Modelle sind genauer. So ist ein 1-MB-Modell genauer als ein 64-kB-Modell, aber auch rechenintensiver. Für eine genaue Erkennung der Wake Words sind große Wake-Word-Modelle erforderlich, damit die Anzahl der Fehl-Erkennungen reduziert wird.

## Audio-Verstärker der Klasse D

Der sprachverarbeitende Teil dieser Steuerschnittstelle wurde stark weiterentwickelt, sodass nun auch preiswerte Geräte eine genaue Spracherkennung bieten können. Die Audioseite der Schnittstelle fand jedoch deutlich weniger Beachtung, weshalb die Klangqualität vieler früherer Smart Speaker und anderer audiofähiger IoT-Geräte im Vergleich zu höherwertigen Audiogeräten minderwertig war.

Vielleicht lenkte die Neuartigkeit der Sprachsteuerung von diesem Schwachpunkt ab. Mit der zunehmenden Verbreitung intelligenter Geräte steigen jedoch auch die Ansprüche der Verbraucher an das damit verbundene Audioerlebnis. Der geringe Wirkungsgrad herkömmlicher Audioverstärker der Klasse AB macht sie für den Einsatz in IoT-Geräten mit geringem Stromverbrauch ungeeignet. Mehrere Chip-Hersteller haben jedoch vor Kurzem eine Reihe hochmoderner Audioverstärker der Klasse D auf den Markt gebracht, die eine deutliche Verbesserung gegenüber den bisher verfügbaren Verstärker darstellen. Viele dieser Verstärker wurden speziell für eine hohe Audioqualität in intelligenten Technologien und IoT-Geräten entwickelt.

Der 15-W-Eingangs-Audioverstärker TAS2770 [1] von Texas Instruments [2] steigert gegenüber herkömmlichen Verstärkern die Lautstärke und die Audioqualität. Zudem sorgt seine verbesserte Spracherkennung für eine einfachere und natürlichere Bedienung von sprachgesteuerten Geräten. Es ist das erste Audio-Front-End, das einen digitalen Mikrofoneingang mit einem leistungsstarken I/V-Sense-Verstärker kombiniert. Dadurch kann es Sprache und Umgebungsgereusche erfassen und gleichzeitig Echo und Rauschen in sprachgesteuerten Anwendungen unterdrücken. Maxim Integrated [3] (jetzt Teil von Analog Devices [4]) hat die Verstärker MAX98357 [5] und MAX98358 [6] der Klasse D mit einem Wirkungsgrad von 92 % und einer Audioleistung ähnlich der der Klasse AB von 3,2 W entwickelt. Ein vereinfachtes Blockschalt-

Bild 2. Vereinfachtes Blockschaltbild der Klasse-D-Audioverstärker von Maxim Integrated. (Quelle: Maxim Integrated)

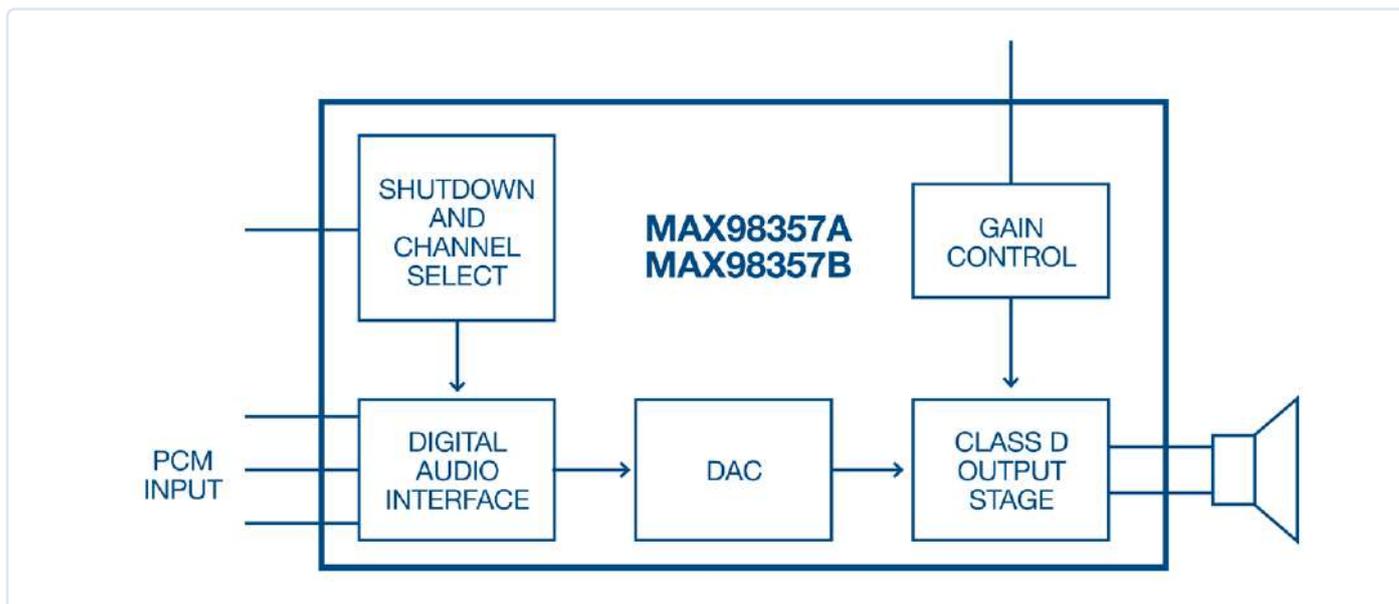


Bild 3. SLN-LOCAL2-IOT Offline-Sprachsteuerungslösung von NXP. (Quelle: NXP [10])

bild für diese Verstärker finden Sie in **Bild 2**. Der PAM8106 [7] von Diodes Incorporated [8] zeichnet sich durch einen geringen Stromverbrauch aus, sodass er sowohl in Geräten mit 1,5-V-Bleisäurezellen als auch mit 3,5-V-Lithium-Ionen-Batterien gut funktioniert. Er hat einen Wirkungsgrad von 92% und kann ohne einen großen Kühlkörper in ein kompaktes Design integriert werden.

### Offline-Sprachsteuerung

Cloudbasierte Lösungen wie Amazons Alexa und Google Assistant eignen sich für Geräte mit einer stabilen Internetverbindung, aber für Geräte mit schlechter oder gar keiner Verbindung ist die Offline-Sprachsteuerung die bessere Lösung.

Wenn ein Produkt beispielsweise auf einfache Ein-Wort-Befehle wie „Go“, „Stop“, „Reset“ (oft als Keyword-Spotting bezeichnet) reagieren soll, ist es sinnvoll, wenn die Verarbeitung lokal auf dem Gerät selbst erfolgt. Die Implementierung eines einfachen Keyword-Befehlssystems ist mit einem kostengünstigen Embedded-Mikrocontroller wie dem EdgeReady MCU-basierten System von NXP für die lokale Offline-Sprachsteuerung möglich. Diese Lösung verwendet die i.MX RT Crossover-MCU und bietet Entwicklern die Möglichkeit, ihre Produkte schnell und unkompliziert mit Sprachsteuerung auszustatten. Die auf i.MX RT106S basierende Lösung von NXP umfasst das in **Bild 3** dargestellte SLN-LOCAL2-IOT-Development Kit [9].

Diese Lösung enthält eine vollständig integrierte Software, die unter FreeRTOS läuft, und wird mit einem Software Development Kit (SDK) geliefert, um einen schnellen Konzeptnachweis zu gewährleisten. Damit können Hersteller problemlos eine Sprachsteuerung in ihre Produkte für Smart Home, Smart Appliance, Smart Building und Smart Industrial integrieren, ohne dass eine WLAN- oder Cloud-Verbindung erforderlich ist. Die Offline-Sprachsteuerung trägt auch den Datenschutzbedenken vieler Verbraucher Rechnung, die befürchten, ihr System könnte durch Online-Eindringlinge gefährdet sein. Das Kit enthält auch ein Windows™-Tool, mit dem Sprachmodelle für über einhundert benutzerdefinierte Befehle und mehrere Wake Words aus Texteingaben in über vierzig Sprachen erstellt werden können.

## Fazit

Sprache und Audiosignale werden in vielen intelligenten Geräten schnell zur bevorzugten Schnittstelle. Sie eignen sich jedoch besonders gut für den Einsatz in stromsparenden tragbaren IoT-Geräten, denn sie machen teure und stromfressende Digitaldisplays

überflüssig. Viele frühe Systeme hatten eine schlechte Audioqualität und konnten nur über eine Cloud-Lösung implementiert werden.

Mit einer neuen Generation hocheffizienter Audioverstärker der Klasse D können die Hersteller den Verbrauchern jetzt allerdings ein hochwertiges Audioerlebnis bieten. Zudem sind jetzt Lösungen erhältlich, mit denen sich auch Geräte steuern lassen, die über eine schlechte oder gar keine Internetverbindung verfügen. Alle diese Innovationen zeigen, dass die Technologie an neue Anforderungen angepasst werden kann, denn die Menschen gewöhnen sich immer mehr an diese Art der Steuerung. Dieser Trend wird sich wahrscheinlich fortsetzen. ◀

220620-02



## Über den Autor

Als Technical Marketing Manager für EMEA bei Mouser Electronics ist Mark Patrick für die Erstellung und Verbreitung von technischen Inhalten in der Region verantwortlich - Inhalte, die für

Mousers Strategie zur Unterstützung, Information und Inspiration der Elektronik-Branche von zentraler Bedeutung sind.

Bevor er das Technische Marketing-Team leitete, war Patrick Teil des EMEA-Lieferanten-Marketing-Teams und spielte eine wichtige Rolle beim Aufbau und der Entwicklung von Beziehungen zu wichtigen Produktionspartnern. Zusätzlich zu einer Vielzahl von technischen und Marketing-Positionen war Patrick acht Jahre lang bei Texas Instruments in den Bereichen Anwendungsunterstützung und technischer Vertrieb tätig.

## WEBLINKS

- [1] TAS2770 Audio Amplifier Evaluation Module : <http://elektor.link/MouserTAS2770>
- [2] Texas Instruments @ Mouser: <http://elektor.link/MouserTexasInstruments>
- [3] Maxim Integrated @ Mouser: <http://elektor.link/MouserMaxim>
- [4] Analog Devices @ Mouser: <http://elektor.link/MouserAnalogDevices>
- [5] MAX98357: <http://elektor.link/MouserMAX98357>
- [6] MAX98358: <http://elektor.link/MouserMAX98358>
- [7] PAM8106 10W Audio Amplifier : <http://elektor.link/MouserPAM8106>
- [8] Diodes Incorporated @ Mouser: <http://elektor.link/MouserDiodesIncorporated>
- [9] SLN-LOCAL2-IOT Solution for Local Voice Control: <http://elektor.link/MouserSLNLOCAL2IOTDevkit>
- [10] SLN-LOCAL2-IOT Product: <http://elektor.link/MouserSLNLOCAL2IOT>

# WEEF 2022

## im Rückblick

Von C. J. Abate (Elektor)

Die Ethikbewegung in der Elektronik stand im Mittelpunkt des zweiten World Ethical Electronics Forum der Lenthe-Stiftung, das auf der electronica 2022 in München stattfand.



Das 2. World Ethical Electronics Forum fand auf der electronica 2022 in München statt. (Quelle: Messe München)

Am 15. November 2022 brachte das World Ethical Electronics Forum (WEEF) auf der electronica in München Ingenieure, Führungskräfte aus der Elektronikindustrie, Akademiker und Technikjournalisten zusammen, um viele der wichtigsten ethischen Fragen der Elektronik unserer Zeit zu diskutieren. Während der eintägigen Veranstaltung boten die Lenthe Foundation - ein gemeinnütziger Ausrichter von Veranstaltungen über soziales Unternehmertum mit Sitz in Amsterdam - und ihre Medienpartner - Elektor, ELEKTRONIKPRAXIS und electronica - eine Plattform für zukunftsorientierte Führungskräfte, um die Themen wie umweltfreundliche Entwicklungspraktiken, die Bedeutung von Zielen für eine nachhaltige Entwicklung, der Planet und die Menschen vor dem Profit, ethische Personalpraktiken, nachhaltige Ressourcen und vieles mehr zu diskutieren.

Die Veranstaltung bot anregende Vorträge und Einblicke von einer Vielzahl von Vordenkern: Prof. Dr. Stefan Heinemann (Professor, FOM Hochschule für Oekonomie und Management), Dr. Reinhard Pfeiffer (Vorsitzender der Geschäftsführung, Messe München), Florian Kurz (KK Ventures), Michel Lorient (AGIGA), Tessa Quandt (VARTA), Dr. Ralf Bodelier (Autor, Journalist), Hendrik-Jan Overmeer (Gründungsmitglied, Deedmob), Andrea Barrett (VP Social Responsibility and Sustainability, RS Group), Robert van Kats (Mitbegründer, bkvv architects), Milda Pladaitė



Der World Ethical Electronics Forum Award 2022. (Quelle: Messe München)



WEEF Award-Gewinnerin Milda Pladaitė - ausgezeichnet von Beatriz Sousa (Elektor). (Quelle: Messe München)



Preisträger Frank Stührenberg (Geschäftsführer Phoenix Contact) mit Dr. Reinhard Pfeiffer (Geschäftsführer der Messe München) und Don Akkermans (Lenthe Foundation / Vorsitzender des Programmausschusses). (Quelle: Messe München)

(Initiatorin von Engineers Enterprise), Kelly Heaton (Künstlerin, USA), Johann Wiesböck (Chefredakteur, Elektronikpraxis), C.J. Abate (Content Director, Elektor), Udo Bormann (Projektleiter des electronica fast forward 2022 Startup Award), Stuart Cording (Journalist, Elektor) und Don Akkermans (Vorsitzender, WEEF). Zusätzlich zu den Vorträgen ehrte die WEEF-Jury vier WEEF-Preisträger:

- > **Sven Krumpel:** Der Inhaber von CODICO, der seinen Mitarbeitern Flächen für den Anbau von Obst und Gemüse zur Verfügung stellt.
- > **Gopal Kumar Mohoto:** Der Ingenieur arbeitet an der Elektrifizierung des Verkehrs in Bangladesch und ist Mitbegründer und CTO von Advanced Dynamics.

- > **Frank Stührenberg:** CEO von Phoenix Contact, Vorstandsmitglied der Stiftung KlimaWirtschaft und einer der Hauptakteure der All Electric Society.
- > **Milda Pladaitė:** Eine Bauingenieurin aus Litauen, die Mitglied des Committee on Young Engineers/Future Leaders bei der World Federation of Engineering Organizations (WFEO) ist.

Die gesamte Veranstaltung wurde von Elektor mit der Kamera festgehalten - sehen Sie es sich auf dem Elektor TV-Kanal unter **elektor.tv** an!



Sind Sie neugierig, wie es mit dem WEEF weitergeht? Besuchen Sie **worldethicalelectronicsforum.com** für aktuelle Nachrichten über das WEEF, den WEEF-Index 2023 und Nachrichten und Artikel über ethische Elektronik. Anfang bis Mitte 2023 wird die Lenthe Foundation (Veranstalter des WEEF) den jährlichen WEEF GUIDE herausgeben. Er wird (online und in gedruckter Form) an Tausende von WEEF-Partnern und Interessengruppen verteilt. Der WEEF GUIDE wird die Profile der im 100 WEEF INDEX aufgeführten Vordenker, Interviews mit Gewinnern, kontroverse Kommentare von außerhalb der Branche, Interviews mit ethisch Verantwortlichen und vieles mehr enthalten. Wenn Sie einen Beitrag leisten möchten, wenden Sie sich bitte an Shenja Panik unter [shenja.panik@elektor.com](mailto:shenja.panik@elektor.com). ◀

(220642-02)RG



Moderator Johann Wiesböck und Don Akkermans im Gespräch mit Wirtschaftsethik-Professor Dr. Heinemann (FOM).



Stuart Cording war einer der beiden Moderatoren, die durch einen Tag voller ethischer Debatten führten. (Quelle: Messe München)



Don Akkermans und C. J. Abate im Gespräch über Forschungsergebnisse zu den Ausstellern der electronica und ihrer ethischen Repräsentation.

# Im Rückblick: FFWD electronica 2022

Innovatoren beeindrucken weiterhin!

Von **Brian Tristam Williams**

Nach Monaten der Vorbereitung und dem Steigen der Spannung wurden die Fast Forward 2022 Awards für Start-ups und Scale-ups, powered by Elektor, auf der electronica 2022 in München verliehen.

Im November, beim electronica Fast Forward 2022 - powered by Elektor, kamen wir wegen der zukünftigen Einhörner und wurden nicht enttäuscht. Endlich konnten wir all die Start-ups und Scale-ups sehen, auf die wir uns in den letzten Monaten gefreut hatten. Am Elektor-Stand haben wir all diese Freunde in einem eigens für den Fast Forward Award 2022 eingerichteten Ausstellungs- und Präsentationsbereich empfangen.

## Die Teilnehmer

Diese Start-ups präsentierten ihre Innovationen an jeweils eigenen Ständen:

- > **Opulo** mit seinem innovativen DIY-Open-Source-Bestückungsautomaten.
- > **AMSEL** gab uns einen detaillierten Einblick in ihre bahnbrechende Lösung zur Oberflächenvorbehandlung.
- > **AirHood** zeigte uns ihre Dunstab-

zugshaube, die keine Abzugshaube ist, sondern eine tragbare, ästhetisch ansprechende Lösung zur Absaugung von Küchendünsten.

- > **V-Juice** präsentierte seine eleganten und schlanken Ideen für kabelloses Aufladen, die mehr als nur Technik bieten – auch in Marketing, Gamifizierung und Incentivierung sind Anwendungen möglich.
- > **Treesense** stellte Umweltsensoren vor, die es Bäumen ermöglichen, mit uns zu sprechen und uns mitzuteilen, was sie benötigen, wodurch viele manuelle Ressourcen bei der Verwaltung und Pflege der Flora eingespart werden können.

Scale-up:

- > **wheel.me**, bereits ein früherer Favorit, stellte seine Lösungen mit autonomen Rollen vor, die mit Sicherheit dem rückenschädigenden Heben und Transportieren in Industrie und Handel ein Ende setzen werden.

Die Innovatoren:

- > **Voltera** stellte seine bestehenden Produkte sowie ein aufregendes neues Produkt vor, mit dem man (fast) alles auf (fast) alles drucken kann.
- > **Quadruped** war ein Favorit der Messebesucher und zog mit seinen vierbeinigen Robotern, die Softwarelösungen des Unternehmens anpriesen, die Massen an.

Und ein Universitätsprojekt:

- > **Ecurie Aix** präsentierte stolz ihr Projekt der RWTH Aachen University, das Jahr für Jahr seine vollautonomen



Der Ausstellungs- und Präsentationsbereich des Fast Forward 2022 Award - powered by Elektor.



Kai Kreos-Nemcock von wheel.me, der Gewinner des ersten Preises.



Die autonomen Rollen von wheel.me erledigen schwere Arbeiten auf intelligente Weise.



Alba Dervishi von Treesense, Gewinner des dritten Preises.

elektrischen Formula-Student-Rennwagen weiterentwickelt.

Alle neun Teilnehmer konnten ihre Entwicklungen am Dienstag und Mittwoch ausführlich vorstellen, und am Donnerstag begeisterten uns die Start-ups und Scale-ups mit ihren offiziellen Quick-Pitches. Die Aufgabe der Jurymitglieder war nicht gerade einfach: Die Beratungen dauerten bis spät in die Nacht, und am Freitag konnten endlich die Gewinner bekanntgegeben werden.

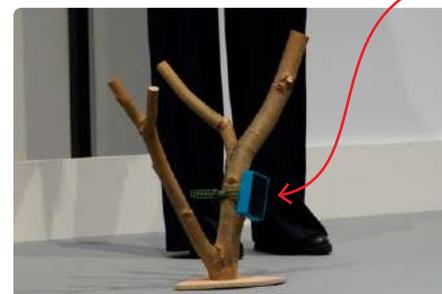
### 1. Preis

Mit einem Medienbudget von 75.000 € von Elektor wurde **wheel.me** ausgezeichnet,

ein Unternehmen, das große Fortschritte bei der Beseitigung schädlicher gemacht hat, mit Anwendungen von der Logistik über die Lagerhaltung bis hin zu Krankenhäusern [1]. Die Rollen von wheel.me sind nicht nur leistungsstark, sondern auch vernetzt, sensibilisiert und autonom. Wir freuen uns darauf zu sehen, wie sie sich weiterentwickeln und wie sie die Industrie beeinflussen werden!

### 2. Preis

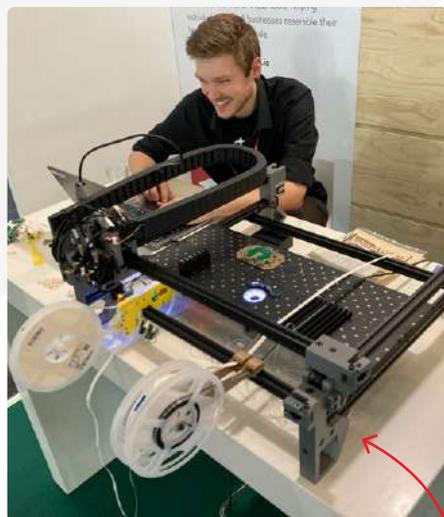
Stephen Hawes war die ganze Woche über mit seiner ansteckender Begeisterung bei der Sache. Er hat seinen Open-Source-Bestückungsautomat **Opulo** Schritt für Schritt zusammengebaut, den gesamten Prozess sorgfältig dokumentiert und auf YouTube „gevlöggt“. Er hat den 2. Preis gewonnen, was bedeutet, dass Stephen die Möglichkeit erhält, sein Produkt anzupreisen und Marketingmittel im Wert von 50.000 € aus dem Elektor Media Kit [2] zu nutzen.



Dank der Treesense-Sensoren können uns die Bäume mitteilen, was sie brauchen.



Der Gewinner des Förderpreises, Dr. Salman Asad von AMSEL.



Stephen Hawes von Opulo richtet die Open-Source-Bestückungsmaschine ein.

### 3. Preis

**Treesense** war der Umweltfavorit unter den Bewerbern, der elektronische Innovationen in Form von eingebetteten Sensoren vorstellte, die den Bäumen den Puls fühlen und den Menschen mitteilen, was sie brauchen [3]. Mit dem Elektor-Medienbudget von 25.000 € kann Treesense ein Thema beleuchten, das noch nicht so viel Aufmerksamkeit erhält, wie es verdient: die Pflege der Bäume auf unserem Planeten.

Ja, wir mussten dieses Jahr einen weiteren Preis erfinden, weil diese innovative Technologie zur Oberflächenvorbehandlung, obwohl sie noch in den Kinderschuhen steckt, nicht unbemerkt bleiben darf. Wir sind gespannt, denn wir haben ihren CEO, Dr. Salman Asad [4], für Fast Forward 2024 wieder eingeladen, wo er beweisen kann, dass da wirklich etwas dran ist.

Herzlichen Glückwunsch an alle Gewinner. Wir werden Ihre Fortschritte in den nächsten Jahren mit großem Interesse verfolgen! ◀

(220651-02)RG

### WEBLINKS

- [1] wheel.me: <https://wheel.me>
- [2] Opulo: <https://opulo.io>
- [3] Treesense: <https://treesense.net>
- [4] AMSELS Salman Asad auf LinkedIn: <https://linkedin.com/in/syed-salman-asad-86337216>



Der Gewinner des zweiten Preises, Stephen Hawes von Opulo.

Das sollte es sein. Von einem 4. Preis war ja nie die Rede. Aber die leidenschaftlich diskutierende, übernächtigte Jury konnte den späten Abend nicht beenden, ohne einen Bewerber zu würdigen, den wir bei der Betrachtung seines Potenzials einfach ermutigen mussten: „Da ist definitiv etwas drin.“

### Förderpreis

AMSEL gewann ein Medienbudget in Höhe von 10.000 €, um seine unglaubliche eBlaze-Technologie auf den Weg zu bringen.

# The Tube

Ein Röhrenverstärker der anderen Art



Von Gerd Reime (Deutschland)

Gute Daten und eine super Optik sind die beiden wichtigsten Aspekte, um die sich dieser besondere Verstärker dreht. Dieses Projekt vereint die handwerkliche Kunst beim Aufbau mit audiophiler Elektronik und verbindet dank Röhrentechnik nostalgische Elemente mit modernem Aussehen. Lesen Sie selbst...

Mit gerade einmal 15 Jahren baute ich 1971 meinen ersten Röhrenverstärker. Es handelte sich um eine Schaltung mit zwei EL84-Röhren im Gegentakt, die ich in Elektor gefunden hatte. Der Rest der Elektronik stammte aus ausgeschlachteten alten Radios. Kurz darauf begann ich eine Lehre als Radio- und Fernsehtechniker bei Grundig. Später arbeitete ich im Bereich Forschung und Entwicklung. Als frisch gebackener Rentner wollte ich mir ein besonderes Projekt vornehmen. Mir schwebte eine kleine, aber feine Endstufe mit etwa 10...15 W Ausgangsleistung vor – mit 2 x EL84, genau wie damals, nur besser.

Zu Anfang wollte ich lediglich ein besonderes Gehäuse haben, aber dann faszinierte mich die Schaltungstechnik und ich versuchte mit vielen Testaufbauten und Messungen etwas möglichst Gutes auf die Beine zu stellen. Aus anfänglichem Spaß wurde ein ernsthaftes Projekt. Wenn Sie sich von meinen Ideen inspirieren lassen wollen, bietet die folgende Beschreibung einen guten Einstieg. Die komplette Baubeschreibung ist ziemlich umfangreich und eigentlich zu groß für einen Elektor-Artikel. Die Details können Sie gerne einer umfassenderen PDF-Datei entnehmen, die unter [1] zum Download angeboten wird.

## Vorüberlegungen

Röhrenverstärker sind immer noch ein Muss für viele Audioenthusiasten. Warum? Angeführt wird häufig der angeblich warme Klang und die Gutmütigkeit bei Übersteuerung. Fast alle mir bekannten Schaltungen arbeiten mit einer Gegenkopplung. Doch welche Auswirkungen hat eine solche Gegenkopplung?

Die Grundaufgabe eines Audioverstärkers ist es, das schwache Signal einer Quelle für die Wiedergabe mit einem Lautsprecher aufzubereiten. Dabei muss sowohl die Spannung als auch der Strom verstärkt werden, damit die praktisch durchweg verwendeten dynamischen Lautsprecher die Spule und somit die Membran weit genug auslenken können, um Musik (oder andere Audiosignale) mehr oder minder laut hörbar zu machen. Bei Verstärkern auf der Basis von Halbleitern ist das kein Problem, doch Röhrenverstärker arbeiten bei verhältnismäßig hohen Spannungen (und entsprechend niedrigen Strömen), die man nicht so einfach auf einen niederohmigen Lautsprecher loslassen kann. Hier ist in der Regel ein Ausgangsübertrager zur Impedanzanpassung notwendig. Über die Frage optimaler Übertrager haben sich schon ganze Generationen von HiFi-Enthusiasten die Köpfe zerbrochen.

Ein weiterer Aspekt ist die Tatsache, dass Röhren im typischen Arbeitsbereich weniger linear arbeiten als es wünschenswert wäre. Damit sich dies nicht allzu verzerrend auf das Audiosignal auswirkt, wird eine Gegenkopplung eingesetzt: Ein Teil des Signals wird vom Ausgang

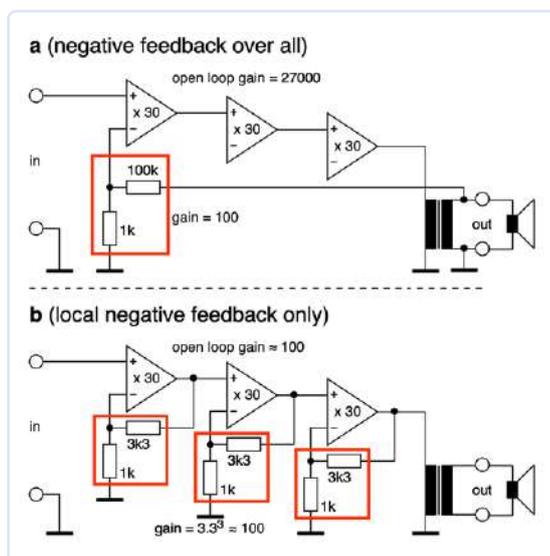
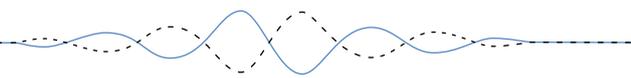
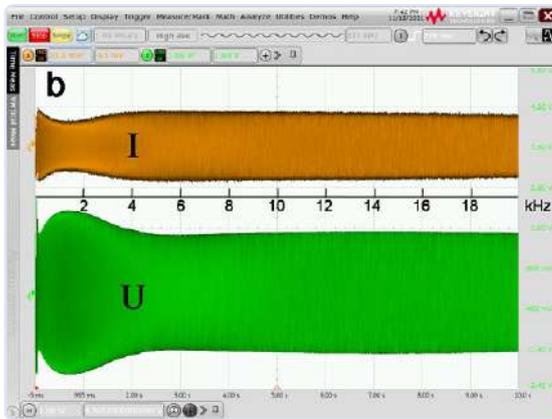
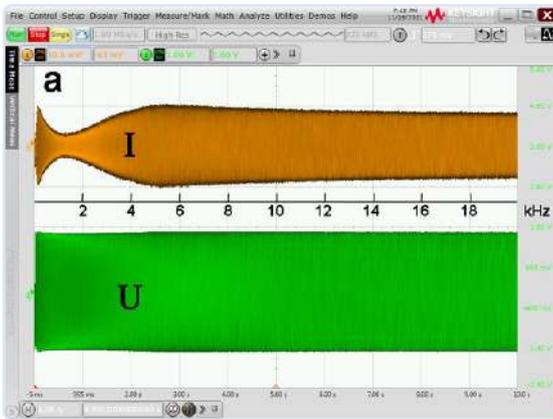


Bild 1. Unterschiedliche Konzepte der Gegenkopplung: a = klassisch, über alles; b = lokal, bei jeder Verstärkerstufe getrennt.

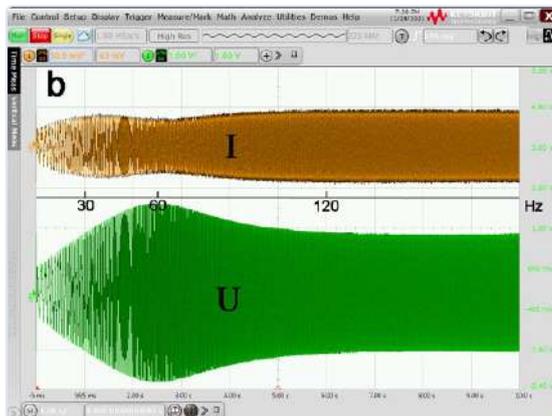
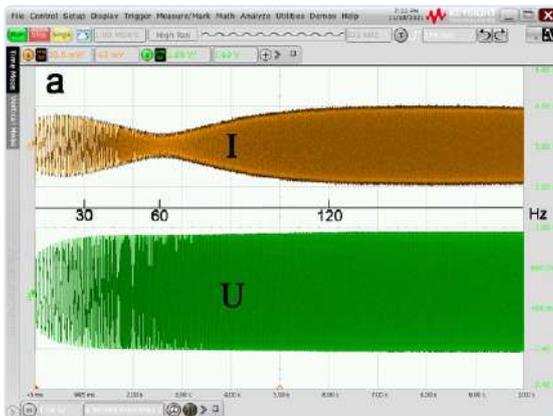


zum Eingang gegengekoppelt und die in der Regel hohe, quasi überschüssige Leerlaufverstärkung (im Verhältnis zum genutzten Verstärkungsfaktor) reduziert dann die sonst auftretenden Nichtlinearitäten. In **Bild 1a** geschieht dies über den Spannungsteiler im roten Rechteck. Alle Nichtlinearitäten und im Verstärker generierten Störsignale werden um einen Faktor reduziert, der dem Quotient aus Leerlaufverstärkung und durch die Gegenkopplung festgelegten Verstärkung entspricht. Im Beispiel nach Bild 1a kommt man auf einen Faktor von  $27.000 / 100 = 270!$  Weil die Gegenkopplung (scheinbar) alles Unangenehme um diesen Faktor reduziert, kommen Unzulänglichkeiten beim Design weniger zum Tragen. Die Alternative zur kräftigen Gegenkopplung „über alles“ (vom Lautsprecherausgang zum Eingang der ersten Verstärkerstufe) besteht darin, die Verstärkung jeder einzelnen Verstärkerstufe quasi durch Gegenkopplung „vor Ort“ zu begrenzen (bei **Bild 1b** die Spannungsteiler in den drei roten Rechtecken, die bei Röhren funktional dem Verhältnis von Anoden- und Kathodenwiderständen entsprechen). Auf diese Weise werden schon die einzelnen Verstärkerstufen „linearisiert“, sodass sich insgesamt ein recht linearer und wenig verzerrender Verstärker ergibt, der nicht komplett „über alles“ gegengekoppelt

werden muss. Die Frage ist, ob das einen Unterschied ausmacht und ob dies tatsächlich anders klingt. Gegenüber einem Transistorverstärker ist der auffälligste Unterschied der, dass ein Ausgangsübertrager vor dem Lautsprecher sitzt. In der Schaltung nach Bild 1a sitzt dieser mit in der Gegenkopplung – bei Bild 1b hingegen nicht. Zu erwarten ist, dass das Konzept nach Bild 1a spannungsmäßig einen recht ausgeglichenen Verlauf aufweist. Meine Messungen an so einem Verstärker in **Bild 2a** bestätigen dies. Das Spektrum der Spannung ist sehr glatt – das des Stroms hingegen weist bei etwa 1 kHz einen Einbruch auf, weil hier beim angeschlossenen Lautsprecher die Impedanz am größten ist. Folglich ist bei 1 kHz auch ein Leistungseinbruch zu verzeichnen. Ein Blick auf **Bild 2b** zeigt, dass aufgrund der größeren Lastimpedanz bei 1 kHz die Spannung steigt und der Stromeinbruch weniger heftig ausfällt. Der Einfluss der Lautsprecherimpedanz auf die abgegebene Leistung wird also zum Teil kompensiert, was einen anderen (besseren) Klang zur Folge haben sollte. Etwas Ähnliches geschieht im Tieftonbereich bei der Resonanz des (Bass-)Lautsprechers rund um die 60-Hz-Marke (**Bild 3**). Auch hier wird der Einfluss der Lautsprecherimpedanz reduziert,

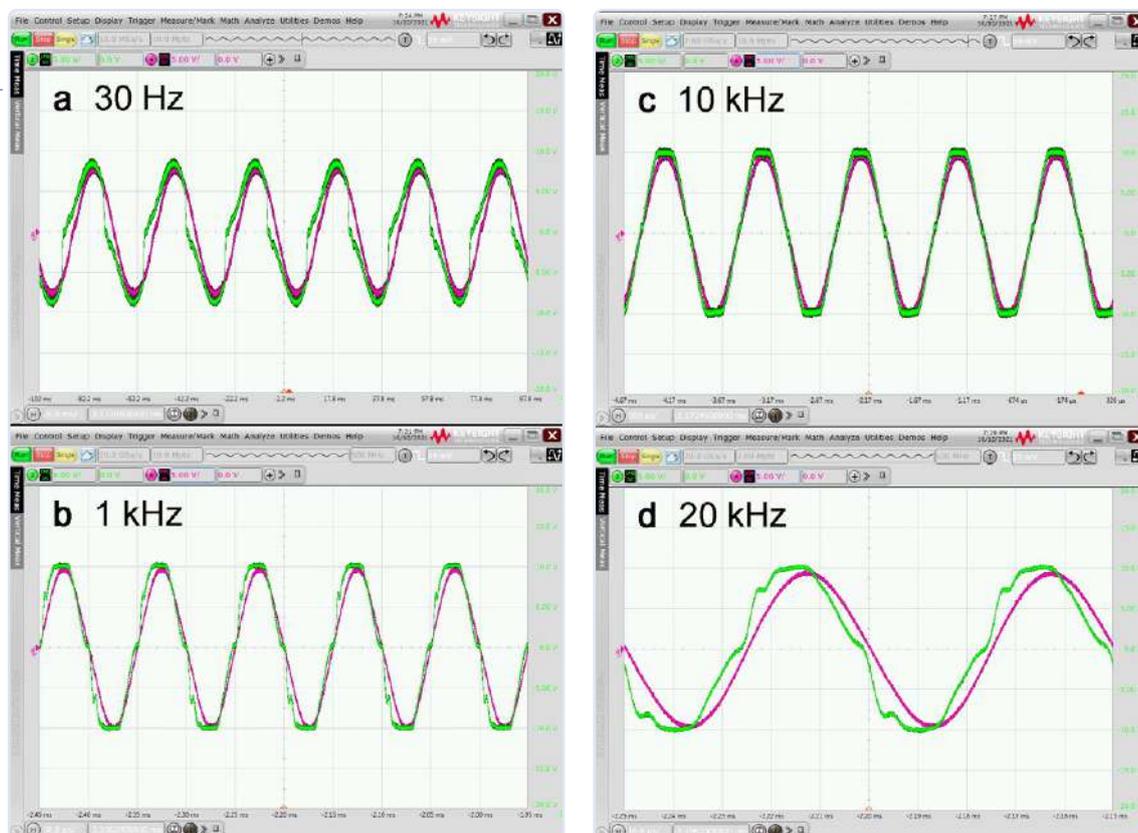


◀ Bild 2. Strom (ocker) und Spannung (grün) über der Frequenz (20 Hz bis 20 kHz): a = mit klassischer Gegenkopplung; b = mit lokaler Gegenkopplung.



◀ Bild 3. Strom (ocker) und Spannung (grün) im Bassbereich. a = mit klassischer Gegenkopplung; b = mit lokaler Gegenkopplung.

► Bild 4.  
Übersteuerungsverhalten  
bei verschiedenen  
Frequenzen (grün =  
klassische und rot =  
lokale Gegenkopplung).



wenn der Ausgangsübertrager nicht im Gegenkopplungs-  
zweig sitzt. Diese Messungen habe ich mit unterschied-  
lichen Boxen wiederholt und es zeigt sich generell ein  
ähnliches Bild.

Ein Röhrenverstärker ohne Gegenkopplung vom  
Lautsprecher zu einer der vorderen Stufen (Variante b)  
verhält sich also deutlich anders als ein „klassischer“  
Verstärker der Variante a. Natürlich habe ich es nicht  
bei reinen Messungen zweier Verstärkerkonzepte belas-  
sen, sondern andere Zeitgenossen um ihren Höreindr-  
uck gebeten. Um es vorwegzunehmen: Gewonnen  
hat bei diesen Vergleichen immer die Variante b ohne  
Gegenkopplung über alles. Der Klang wurde als deutlich  
„wärmer“ und ausgewogener empfunden. Je nach darge-  
botener Musik-Gattung wurde der „Druck“ als intensi-  
ver empfunden. Beides ist durch die unterschiedlichen  
Messkurven nicht weiter verwunderlich.

Noch einige Unterschiede: Bei geringen Leistungen be-  
ziehungsweise Lautstärken arbeitet eine Gegentakt-AB-  
Endstufe ja im A-Betrieb und hat dadurch naturgemäß  
geringe Verzerrungen. Daher sollte der Klirrfaktor bei  
einem guten Ausgangsübertrager bei kleinen Lautstär-  
ken mit und ohne Gegenkopplung nahezu gleich sein.  
Erstaunlicherweise zeigte das Konzept ohne Gegen-  
kopplung über alles im A-Betrieb sogar einen deutlich  
geringeren Oberwellenanteil. Die besten Ausgangsüber-  
trager fand ich übrigens bei Menno Van der Veen [2].

### Clipping und Sättigung

Es ist schon fast eine Binse, dass Röhrenverstärker bei  
hohen Lautstärken, wenn die Spitzenwerte des Signals  
in die Nähe der Versorgungsspannungen kommen, sanfter

begrenzen. In der Regel begrenzen die Endröhren die  
Amplitude – zusammen mit dem Ausgangsübertrager,  
der dann in die Sättigung gerät. Dann wird eine Gegen-  
kopplung über alles aber problematisch, denn genau  
diese Gegenkopplung führt dazu, dass die nichtbe-  
grenzenden Vorstufen beim Clipping der Leistungsstufe  
stärker mit dem Fehlersignal angesteuert werden, was  
einen direkteren und heftigeren Anstieg des Klirrfaktors  
zur Folge hat. Ohne eine solche Gegenkopplung ist das  
Klirrvverhalten bei Übersteuerung deutlich gutmütiger.  
Gerade bei tiefen Frequenzen im Bereich von 20...30 Hz  
können Ausgangsübertrager oft keine großen Leistungen  
mehr übertragen. Die Folge ist, dass die negativen  
Effekte der Gegenkopplung im Tiefbassbereich schon bei  
geringeren Leistungen hörbar werden. Hinzu kommen  
Effekte des Phasenverhaltens des Übertragers, die ja  
ebenfalls durch die Gegenkopplung unerwünschte  
Klangeffekte produzieren.

Meine Messungen bei einer Übersteuerung um 10 %  
und verschiedenen Frequenzen stützen die Vorüberle-  
gungen. Die roten Kurven des Verstärkers ohne globale  
Gegenkopplung in **Bild 4** sehen durchweg „besser“  
aus als die mehr oder minder verzerrten grünen  
Signale des voll gegengekoppelten Verstärkers. Be-  
sonders extrem fallen die Unterschiede bei 20 kHz  
aus, doch auch bei 30 Hz zeigen sich relevante  
Verschlechterungen des klassischen Verstärker-  
Designs. Interessanterweise tritt die zehnpromen-  
tliche Übersteuerung bei dieser Frequenz schon bei  
85 % des Nennpegels auf. Zusammenfassend lässt  
sich mit Recht behaupten, dass das Clipping nur  
ohne globale Gegenkopplung wirklich weich und  
kaum störend auftritt.

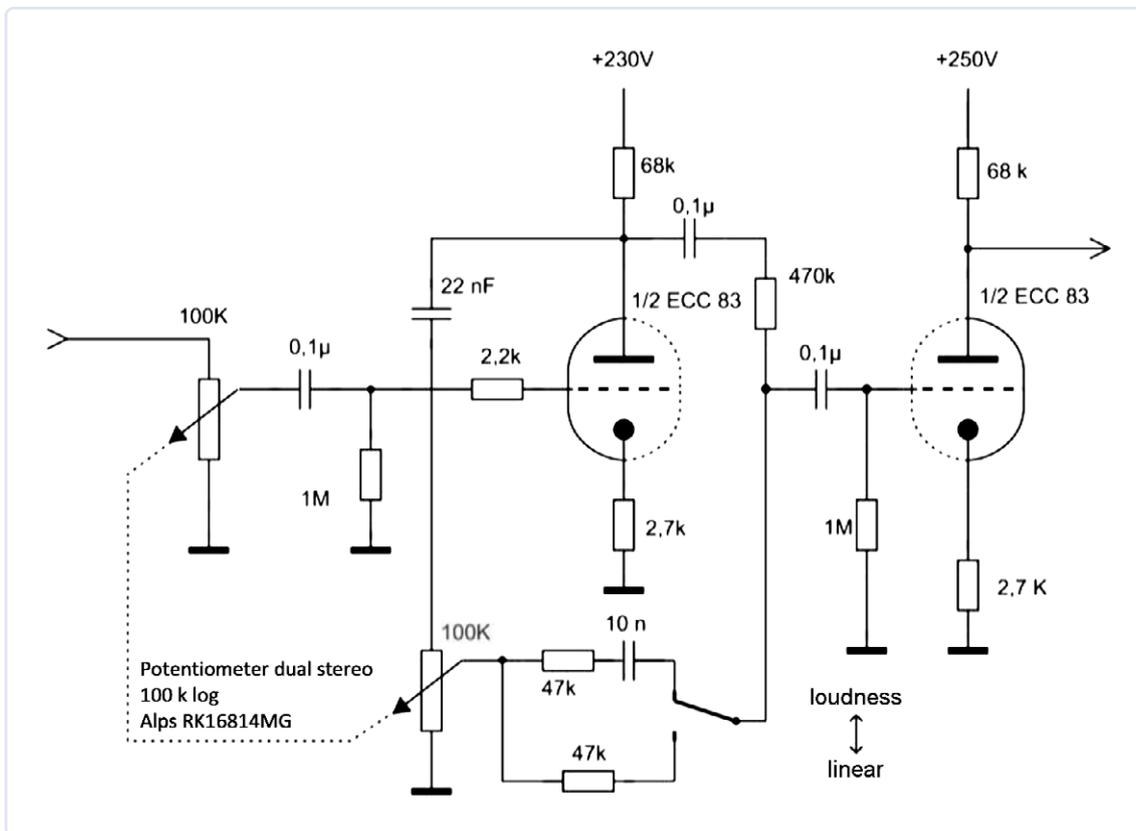


Bild 5. Schaltung der gehörrichtigen Lautstärkeregelung mit Vierfach-Poti von Alps.

## Gehörrichtige Lautstärkeregelung

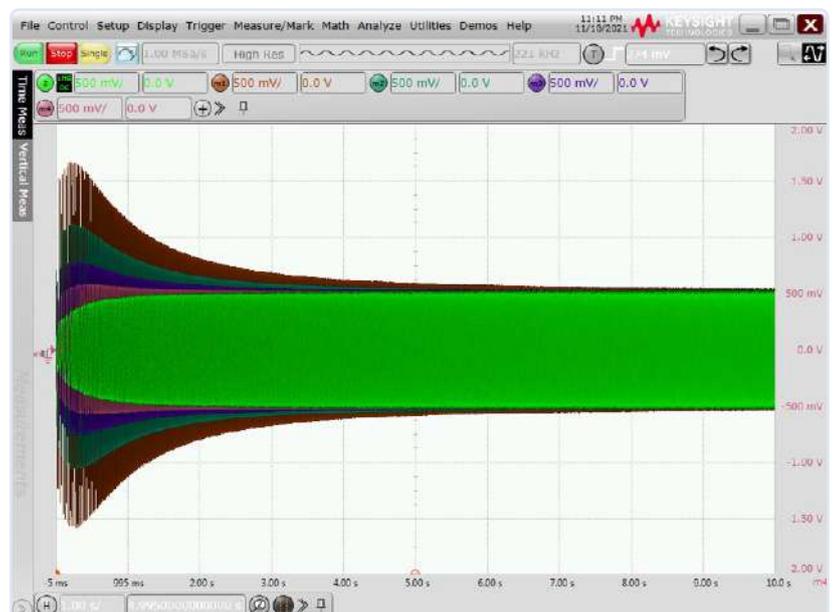
Häufig liest man die Behauptung, dass ein linearer Frequenzgang das einzig Wahre sei. Demnach müssen hochwertige Verstärker einen flachen Verlauf des übertragenen Frequenzspektrums aufweisen. Ich hingegen hege den Verdacht, dass die Forderung nach flachen Frequenzgängen lediglich daher rührt, dass heute Volume-Potis mit Anzapfungen nicht mehr erhältlich sind. Im Netz zu findende Ersatzschaltungen funktionieren nur so halbwegs. Hintergrund ist, dass die menschlichen Ohren bei geringen Lautstärken bei Höhen und Tiefen weniger empfindlich sind – oder aber die Mitten bevorzugen – ganz wie man die Sache sieht.

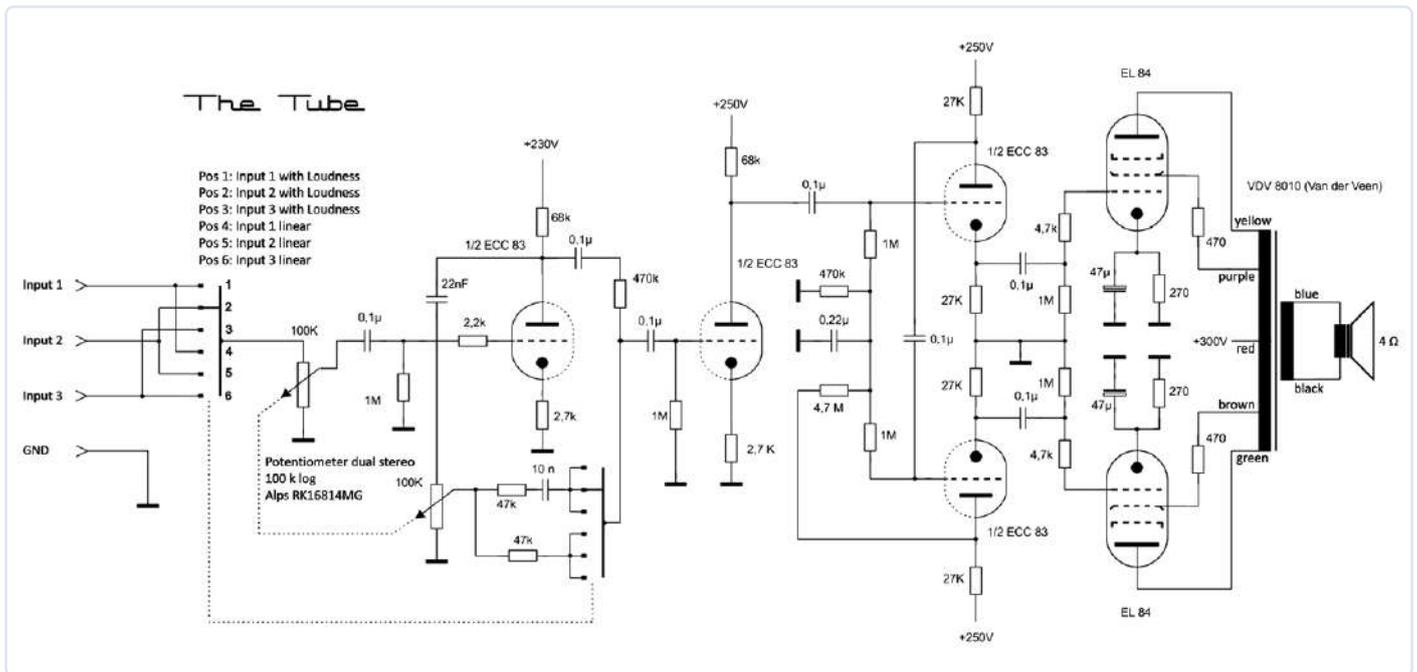
Früher waren sogenannte gehörrichtige oder auch physiologische Lautstärkereglern (der englische Begriff lautet „loudness“) der übliche Standard. Dabei wurden bei geringer Lautstärke Höhen und Tiefen angehoben und diese Anhebung ging mit zunehmender Lautstärke zurück. Aber welcher moderne Verstärker kann heute noch mit so einer Funktion aufwarten? So gut wie keiner! Statt Vintage-Potis mit zwei oder drei Anzapfungen, die aufgrund der Stufen sowieso nicht ideal waren, kann man das Problem auch anders und eleganter lösen: Man nehme einen Dual-Stereo-Poti, also faktisch ein Vierfach-Potentiometer, wie es zum Beispiel von Alps nach wie vor gefertigt wird. Damit und einer einfachen Beschaltung lässt sich ein hervorragender Lautstärkereglern mit gehörrichtiger Lautstärkereglern aufbauen. Außerdem kann man dabei mit einem einfachen Umschalter zwischen gehörrichtig und linear ganz ohne Lautstärkeveränderung umschalten.

Bild 5 zeigt die von mir gebaute Schaltung mit dem

Poti Alps RK16814MG. Dabei handelt es sich zwar um ein Motorpoti, doch der Motor kann bei Bedarf entfernt und das Poti händisch bedient werden. Um den Effekt der gehörrichtigen Lautstärkereinstellung zu verdeutlichen, habe ich die Frequenzgänge bei verschiedenen Einstellungen gemessen und nachträglich auf gleiche Pegel bei 1 kHz normiert und die Kurven überlagert. In Bild 6 sieht man den Effekt im Bassbereich. Es wird auch deutlich, dass die Frequenzgangkorrektur mit steigender Lautstärke stetig abnimmt.

Bild 6. Effekt der gehörrichtigen Lautstärkeregelung; **braun** = sehr leise; **dunkelgrün** = leise, Hintergrundmusik; **blau** = mittlere Lautstärke; **rot** = ziemlich laut (Frequenzgang linear); **grün** = volle Lautstärke, kurz vor der Übersteuerung.





**Bild 7.** Die Schaltung eines Verstärkerkanals von The Tube. Besonders interessant ist die gehörrichtige Lautstärkeregelung und die symmetrische Ansteuerung der Endröhren im Gegentakt.

### Ansteuerung der Endröhren

Push-Pull- oder Gegentaktstufen benötigen bekanntlich eine Phasendrehung um 180°. Gegenüber einfachen Schaltungen mit nur einer Röhre kann man mit etwas mehr Aufwand auch mehr Qualität erreichen. **Bild 7** zeigt die Schaltung eines Kanals von „The Tube“. Die beiden als Endröhren fungierenden EL84 werden über jeweils einen eigenen Kathodenfolger aus je 1/2 ECC83 angesteuert. Durch gleiche Anoden- und Kathodenwiderstände kann die obere Triode präzise das Gitter der unteren Triode steuern und so eine fast perfekte, symmetrisch konfigurierte Phasendrehung realisieren. Dieser Aufwand reduziert Verzerrungen und Phasenfehler. Auf

diese Weise entsteht auch ohne Gegenkopplung über alles ein ordentlicher Verstärker mit typischem Röhrenklang, aber ohne erhöhten Klirrfaktor. Diverse Hörvergleiche bestätigen, dass dieses Konzept (subjektiv) sogar besser als ein hochwertiger, sehr teuer und wenig energieeffizienter Röhrenverstärker in Klasse-A-Betrieb ist und sogar gute Transistorverstärker in die Schranken verweisen kann.

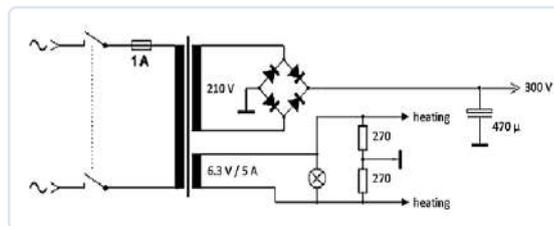
In **Bild 8** ist das recht einfach konzipierte Grundschaltung des Netzteils zu sehen. Details sind unter [1] zu finden.

### Making-of

Alles fing damit an, dass meine Söhne irgendwoher größere Metallrohre organisierten. Und aus solchen Rohren kann man eigentlich nur eines machen, dachte ich: einen Röhrenverstärker!

Ein massiver Alu-Deckel dient als Basis für schöne Röhrensockel. Wie in **Bild 9** zu sehen, stehen End- und Vorstufenröhren eng beieinander. Aufgrund der zu erwartenden thermischen Belastung dürfen hier keine Kunststoffteile zum Einsatz kommen.

Die Unterseite dieser Alu-Platte (**Bild 10**) ist mit einigen



**Bild 8.** Die Grundschaltung des Netzteils ist recht einfach.



**Bild 10.** Selbst die Abstandshalter sehen auf einer massiven Alu-Platte gut aus.

**Bild 9.** So ästhetisch kann die zur Elektronik gehörende Mechanik ausfallen: Die Sockel für alle Röhren des Stereo-Verstärkers.

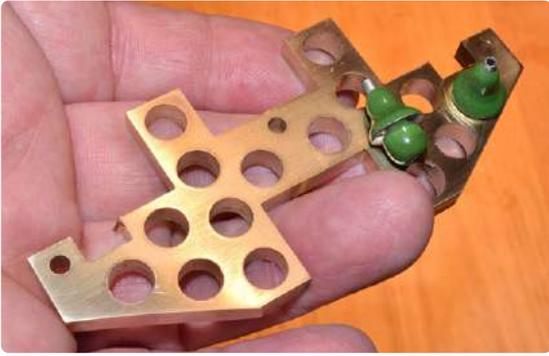


Bild 11. Voll adäquat für Röhrentechnik: Eine „Platine“ aus massivem Messing.



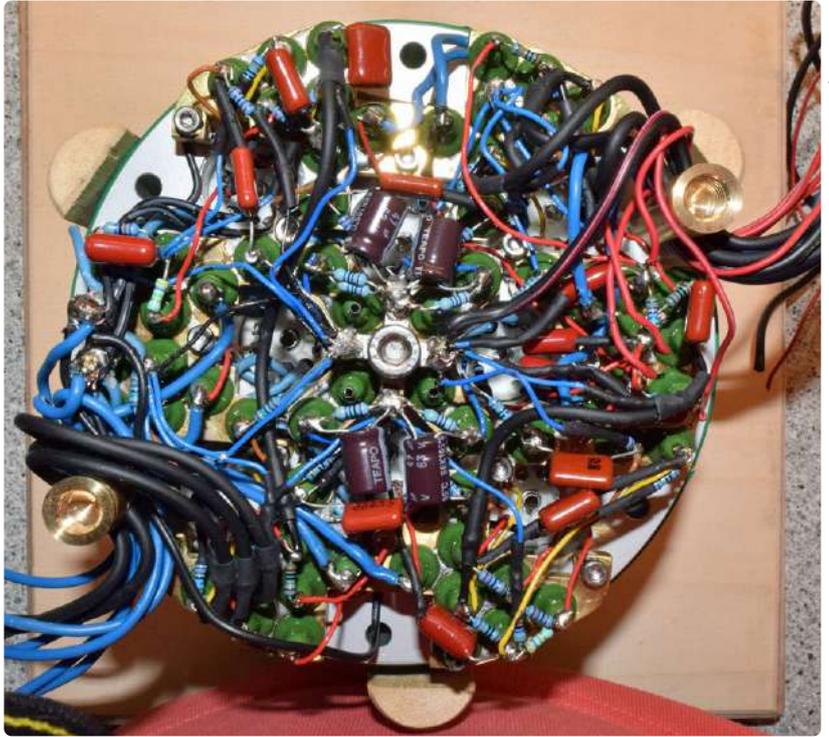
Bild 12. Die „Löt-Pads“ mit fast antiken, grünen Glasisolatoren.

Abstandshaltern versehen, auf denen die „Platine“ verschraubt wird. Platine steht zurecht in Anführungszeichen, denn erstens erwartet man so etwas nicht unbedingt bei einem Röhrenverstärker und zweitens kommt hier kein Epoxid-Material zu Einsatz, sondern aus massiven Messingplatten gefräste „Bauteilehalter“ (Bild 11). So ein Röhrenverstärker ist auch inwendig immer etwas fürs Auge, nicht wahr? Bild 12 zeigt die Unterseite der Aluplatte mit fertiger Messingplatine und ihren schönen „Löt-Pads“ mit grünen Glasisolatoren.

Fertig „bestückt“ sieht diese Platine so aus wie in Bild 13.



Bild 14. Dicke Elkos brauchen eine eigene Messingplatine.



Eine weitere „Platine“ trägt größere Bauteile wie die Elkos (Bild 14). Auch für den Netztrafo – aufgrund der runden Form natürlich eine Ringkernauführung – wurden tragende Messingplatten angefertigt, die in Bild 15 zu bewundern sind. Das war selbstverständlich noch längst nicht alles an Mechanik. Die Frontplatte besteht logischerweise aus einem Ring aus Alu, in den Eingangswahlschalter, Netzschalter und Lautstärke-Poti eingebaut



Bild 13. Die fertig bestückte Platine von der kombinierten Bauteil-/Lötseite aus gesehen. Nicht sehr übersichtlich, aber voll stabil.



Bild 15. Ein runder Trafo für einen runden Verstärker. Auch hier ist extra Mechanik zur Befestigung erforderlich.



►  
Bild 16. Die Bedienelemente Wahlschalter, Netzschalter und Lautstärke-Poti sitzen in einem eigenen Alu-Ring.



Bild 17. Der komplette Verstärker provisorisch zusammengebaut, noch ohne schützendes, röhrenförmiges Gehäuse.



Bild 18. Der fertige Verstärker in voller Pracht. Eher ein Schmuckstück als bloß wohnzimmer-kompatibel.

sind (Bild 16). Eine schönen Eindruck dieser Komposition zeigt der provisorisch zusammengebaute Stapel von Bild 17. Und nun – Tadah! – der fertige Verstärker im Gehäuse (Bild 18).

So ein Röhrenverstärker ist bei entsprechender Liebe zum Detail immer ein besonders ästhetisches Stück Technik. Doch fehlt da nicht noch etwas Entscheidendes? Selbstverständlich müssen auch bei The Tube schwache Audio-signale rein und kräftige Signale raus zu den Lautsprechern kommen. Irgendwie muss auch Netzspannung an den eingebauten Trafo gelangen. Die Lösung für dieses



Bild 19. Ein Ausschnitt auf der Rückseite birgt Ein- und Ausgänge sowie eine Netzbuchse.

Problem ist in Bild 19 zu sehen: Dieser Ausschnitt im röhrenförmigen Röhrenverstärkergehäuse beherbergt die vergoldeten Cinch-Buchsen für die Eingänge, XLR-Buchsen für die Lautsprecher und eine Kaltgerätebuchse für das Netzkabel.

### Außerdem

Mein Projekt dürfte klar machen, dass es bei einem Röhrenverstärker mit einer guten Schaltung noch lange nicht getan ist. Um Thomas A. Edison zu paraphrasieren: Neben dieser Inspiration (Elektronik) braucht es noch sehr viel Transpiration (Mechanik). Wenn Sie sich also von meinem Projekt inspirieren lassen, sollten Sie bereit sein, viel Arbeit in den Aufbau zu stecken, um nicht nur einen gut klingenden, sondern auch optisch ansprechenden Röhrenverstärker zu realisieren. Das alte Sprichwort „ohne Fleiß kein Preis“ gilt in der Röhrentechnik in besonderem Maße. ◀

220089-02

### Über den Autor

Gerd Reime arbeitete nach einer Ausbildung als Radio- und Fernsehtechniker in den Entwicklungsabteilungen von zunächst Grundig und später Nokia. Er ist Inhaber mehrerer Patente.

### Sie haben Fragen oder Kommentare?

Bei technischen Fragen können Sie sich gerne an die Elektor-Redaktion wenden. Die E-Mail-Adresse lautet: [redaktion@elektor.de](mailto:redaktion@elektor.de).



### Passende Produkte

- **Menno van der Veen, Trans-Röhrenverstärker**  
E-Buch (PDF): [www.elektor.de/17346](http://www.elektor.de/17346)
- **Menno van der Veen, Der Entwurf von Röhrenverstärkern**  
E-Buch (PDF): [www.elektor.de/17372](http://www.elektor.de/17372)

### WEBLINKS

- [1] Umfangreiche Projektbeschreibung im PDF: <https://www.elektormagazine.de/220089-02>
- [2] Übertrager von Vanderveen: <https://www.mennovanderveen.nl>
- [3] Firma Picosens: <https://picosens.de>

# Biomaterialien in der Elektronik: Bereit oder nicht?



Source: CHUTTERSMAP, Unsplash

Von Priscilla Haring-Kuipers (Niederlande)

Ein Großteil der Defekte in der Elektronik geht auf die verwendeten Materialien zurück. Sind wir bereit für einen Wechsel zu umweltfreundlicheren Biomaterialien in der Elektronikfertigung?

## Recycling

Nachhaltigkeit im Bereich der Elektronik bezieht sich in der Regel auf das Recycling, und in diesem Bereich gibt es in der Tat noch viel Handlungsbedarf. Nicht einmal 20 % der Elektronik werden weltweit recycelt, und davon wird das meiste von undurchsichtigen Elektronik-Recyclingunternehmen oder sogar von Urban Miners und Schrotthändlern erledigt [1]. Aber auch von der anderen Seite der Elektronikherstellung lässt sich die Situation verbessern. Indem wir

das Ausgangsmaterial verändern, können wir auch den Recyclingprozess verändern. Mit Blick auf die Kreislaufwirtschaft könnten Biomaterialien entwickelt werden, die sauber, ungiftig und leichter zu verarbeiten sind.

## Das Problem mit FR4

Fast alle elektronischen Geräte werden auf Platinen aus FR4 und Kupferfolie aufgebaut. FR steht für Flammschutzklasse (flame retardant class) und wird durch Eintauchen von Glasfasergewebe in ein Epoxidharz hergestellt. Epoxidharz ist ein duroplastisches Polymer mit einem „Epoxidring“ irgendwo in der Molekularstruktur [2]. Epoxidharz stammt in der Regel aus der Petrochemie und hat einen großen CO<sub>2</sub>-Fußabdruck (5,7...7,6 kg CO<sub>2</sub> pro Kilogramm) [3]. Zudem ist die chemische Verarbeitung von Epoxidharz eine bekannte Ursache für berufsbedingtes Asthma, das die Lunge dauerhaft schädigen kann.

Die hohe Viskosität des Harzes bei Raumtemperatur erfordert außerdem die Zugabe von petrochemischen Verdünnungsmitteln.

Epoxidharz ist ein duroplastischer Kunststoff, was bedeutet, dass er sich nicht in seine chemischen Bestandteile zerlegen und sauber recyceln lässt, so wie ein gebackener Kuchen nicht in Mehl, Eier und Butter zurückverwandelt werden kann. Die chemische Reaktion, die diesem Material fantastische mechanische, dialektische, flammhemmende sowie hitze- und feuchtigkeitsbeständige Eigenschaften verleiht, führt auch dazu, dass es sehr schwer zu recyceln ist. „Nachhaltige Kreislaufwirtschaft erfordert mehr Forschung, um leistungsstarke biobasierte Polymere zu finden, die synthetische Polymere in Platinen ersetzen können“ [4].

Die Herstellung, Verarbeitung und Entsorgung von FR4 bringt viele Probleme mit sich, die durch die Verwendung anderer Materialien gelöst werden könnten. Einer der Bestandteile von FR4 war früher Steinkohlenteer, aber wir haben es durch andere Derivate fossiler Brennstoffe ersetzt. Es sieht so aus, als ob wir bereit sind, zusätzliche Inhaltsstoffe zu entwickeln.

Bei einigen Projekten wurden Papier, Flachfasern und biobasierte Harzmatrizen verwendet, wobei weniger giftige Flammschutzmittel oder Ersatzstoffe auf Soja- und Hanfölbasis eingesetzt wurden. Die Haltbarkeit ist bei diesen Bioplatinen ein Problem, da sie - im Gegensatz zu FR4 - mit Feuchtigkeit reagieren.

## Mehr Bio

Polymilchsäure (polylactic acid, PLA) ist der bekannteste Biokunststoff, der vollständig aus erneuerbaren Rohstoffen hergestellt wird. Leider verträgt er keine Hitze, und die Prozesse der Elektronikherstellung stellen oft hohe Temperaturanforderungen an das Material (Aufdampfen von Metallen, Reflow-Öfen oder Wellenlöten). Es wird aber eine zusätzliche chemische Verarbeitung von PLA entwickelt, um es hitzebeständiger zu machen. Wie mir ein Chemiker kürzlich sagte: „Wir können jetzt alle Arten von Biokunststoffen herstellen. Sie brauchen nur Ihre Anforderungen zu nennen.“

Leitfähiges Material auf Kohlenstoffbasis ist hitzebeständiger und hat eine bessere elektrische Leitfähigkeit. Es gibt eine ganze Reihe potenziell relevanter leitfähiger Kohlenstoffe, die alle aus erneuerbaren Rohstoffen synthetisiert werden und die Verwendung verschiedener Metalle in der Elektronik ersetzen könnten. Kohlenstoff-Nanoröhren können aus praktisch jeder organischen Verbindung gefertigt werden, und Graphen kann durch Exfoliation von Graphit hergestellt werden. Mehrere Universitäten in den USA arbeiten an biobasierter Graphen-Tinte für den Druck von elektronischen „Herstellungswerkzeugen, die derzeit nur im Labor oder in der Fantasie existieren“, sowie an biobasierten Substraten, auf die sie gedruckt werden können. Mehrere finnische Universitäten arbeiten an der Herstellung flexibler piezoelektrischer Sensoren aus



Source: Michael Schiffer, Unsplash

Holzfasern und Bakterien sowie an der Herstellung von Superkondensatoren aus Löwenzahn. Sie entwickeln „neue grüne Materialien“ für Sensoren, Elektroden, Platinen und Bauteile, die alle auf Zellulose und anderen biobasierten Materialien beruhen.

Aber auch die Definition von „biobasiert“ entwickelt sich weiter. Nach Angaben der amerikanischen Food and Drug Administration (FDA) umfasst der Begriff „Biokunststoffe“ jetzt auch Produkte, die aus erdölbasierten Polymeren hergestellt und mit Naturfasern gemischt werden. Solche Materialien wären nicht biologisch abbaubar, und das darf nicht das sein, was uns vorschwebt, wenn wir an Biokunststoff denken.

### Langlebig

Die Probleme von petrochemischen Kunststoffen und Biomaterialien für die Elektronik sind zwei Seiten derselben Medaille: extrem haltbar versus biologisch abbaubar. Petrochemische Materialien, die in der Elektronik verwendet werden, werden durch natürliche Alterungsfaktoren wie Feuchtigkeit und Temperatur nicht sehr beeinträchtigt. Sie sind so haltbar, dass sie sich überhaupt nicht zersetzen und ewige Berge von Elektroschrott bilden, den wir nicht richtig recyceln. Biomaterialien bauen sich mit der Zeit ab, was zwar gut klingt, wenn ein Gerät am Ende seiner Lebensdauer angelangt ist, aber weniger gut, wenn man noch dabei ist, das Gerät zu entwerfen.

Eigentlich wollen wir, dass die meisten unserer elektronischen Geräte sich abbauen lassen. Einige Geräte sind ohnehin nur für den einmaligen Gebrauch bestimmt, wie RFID-Tags, Sensoren und medizinische Kits. Auch die meisten anderen Geräte wollen wir nicht jahrzehntelang - oder nur ein Jahrzehnt - verwenden. Vielleicht sollte nicht unbegrenzte Haltbarkeit, sondern biologisch abbaubare Elektronik der Standard bei der Herstellung sein. Wenn die Industrie *wirklich* Kunststoffe oder -harze verwenden muss, die ewig halten, sollte sie auch einen Plan vorlegen, wie sie für Ihr wertvolles elektronisches Gerät mindestens die nächsten 50 Jahre Support leisten will.

Der verantwortungsvolle Umgang mit Werkstoffen ist im Anmarsch und wird in politischen Maßnahmen wie dem Green Deal der EU und dem aktuell diskutier-

ten Climate Law der USA und festgeschrieben. Es gibt einige fantastische Entwicklungen in der Materialwissenschaft, die darauf abzielen, die „Verwirklichung einer biobasierten Wirtschaft zu beschleunigen“ und die für die Herstellung von Elektronik relevant sind. Aber soweit ich weiß, ist noch nichts davon in großem Maßstab verfügbar. Die meisten Forschungsprojekte scheinen jedoch bereit zu sein, in den nächsten fünf Jahren in den Chemie- und/oder Produktionsmarkt einzusteigen. Ich erwarte große Bio-Dinge! ◀

(220556-02)SG

Source: Fukayamamo, UnSplash



## WORLD ETHICAL ELECTRONICS FORUM

### Weltforum für ethische Elektronik 2023

Im November 2021 startete Elektor das **World Ethical Electronics Forum** (WEEF) in München. Die offene Diskussion über Ethik und nachhaltige Entwicklungsziele (SDGs) der Veranstaltung regte globale Innovatoren der Elektronikbranche an. Neben Elektor-Ingenieuren und -Redakteuren gehörten Dr. Stefan Heinemann (Professor für Wirtschaftsethik an der FOM Hochschule), Dr. Paula Palade (PhD, Jaguar Land Rover), Margot Cooijmans (Direktorin, Philips Foundation) und mehrere andere Vordenker, darunter Priscilla Haring-Kuipers, zu den Rednern und Diskussionsteilnehmern. Besuchen Sie die WEEF-Webseite, um auf dem Laufenden zu bleiben und sich zu engagieren: [www.worldethicalelectronicsforum.com](http://www.worldethicalelectronicsforum.com).

### WEBLINKS

- [1] eWaste Ben: <https://www.youtube.com/c/eWasteBen>
- [2] Custom Materials, Inc, „Is Epoxy Resin Plastic?“, 2. Januar 2022: <https://www.youtube.com/watch?v=i8nEQQ9S0V0>
- [3] NIH, „Biolastics and Carbon-Based Sustainable Materials, Components, and Devices: Toward Green Electronics“, 20. Oktober 2021: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8532127/>
- [4] O. Ogunseitun, et al, „Biobased materials for sustainable printed circuit boards“, Nature Reviews, 12. September 2022: <https://www.nature.com/articles/s41578-022-00485-2#citeas>

# Opera-Cake-Antennenumschalter für HackRF One

Schließen Sie bis zu acht Antennen an Ihr SDR an

Von Clemens Valens (Elektor)

Opera Cake von Great Scott Gadgets ist ein Antennenumschalter für das HackRF-One-SDR. Mit der Platine können Sie bis zu acht Antennen an das System anschließen oder eine schaltbare Filterbank realisieren.



Operntorte von la-fontaine, Pixabay

In der Tradition elektronischer Geräte, die nach Kuchen, Gebäck und Süßigkeiten benannt sind (man denke an Raspberry Pi und Banana Pi, Snickerdoodle (Zimtküchlein) [1] und frühe Versionen von Android), ist auch der Opera-Cake-Antennenumschalter ein Gerät, das in keinerlei Beziehung zu seinem Namen steht (zugegeben, Pi bezieht sich wenigstens auf Python).

Die Operntorte, die ich kenne und sogar ein paar Mal gebacken habe, ist ein mehrschichtiger Kuchen, der hauptsächlich nach Mandeln und Kaffeechocolate schmeckt. Vielleicht ist das der Bezug zu dem unten beschriebenen Gerät? Sie werden nämlich eine Menge Kaffee brauchen, um während der langen Stunden des SDR-Spaßes, die es Ihnen bescheren wird, wach zu bleiben.

## Der Opera-Cake-Antennenumschalter ist ...

Wie bereits erwähnt, ist der Opera Cake ein Antennenumschalter für den HackRF One [2], wie in **Bild 1** gezeigt. Eigentlich handelt es sich um einen doppelten Vierfach-Antennenschalter, das heißt, um ein Paar einpolige Vierfachschalter (SP4T). Sie können die beiden Schalter - auch als Bänke bezeichnet - parallel als DP4T-Schalter oder in Reihe (sozusagen) als SP8T-Schalter nutzen. Die Schalter sind bidirektional, funktionieren also sowohl im RX- als auch im TX-Modus.

Beginnen wir mit der Beschreibung, wie Great Scott Gadgets' Version des Opera Cake aussieht. Im Grunde handelt es sich um eine 120 mm x 75 mm große vierlagige Platine (Moment mal, vier Lagen, geschichteter Kuchen, hmm...) mit einer Vielzahl von Anschlüssen: Es gibt fünf SMA-Buchsen an beiden Enden der Platine und drei „stapelbare“ Buchsenleisten, die auf der Platine montiert sind, und Footprints für drei weitere.

## ... eine Aufsteckplatine

Die Platine ist von erstaunlich wenigen elektronischen Bauteilen bevölkert. Das liegt daran, dass Opera Cake eine Zusatzplatine ist. Sie benötigt keine Intelligenz, da sie von der Platine, auf die sie gesteckt wird, gesteuert wird. Von der Größe her ist sie zum Aufstecken auf eine HackRF-One-Platine gedacht. Aber wenn Sie die richtigen Anschlüsse auf die Platine löten, können Sie verkehrt herum auch einen GreatFET One [3] darauf stecken.

Wenn Sie einen HackRF One im Gehäuse haben, müssen Sie das Gerät zuerst (mit Vorsicht) davon befreien. Erst dann können Sie den Opera Cake aufstecken.

## ... konfigurierbar

Die erste Anwendung des Antennenumschalters, die einem in den Sinn kommt, ist natürlich der Anschluss verschiedener Antennen an einen einzigen HackRF One, so dass man nicht jedes Mal sein Setup neu verkabeln muss, wenn man das HF-Band wechseln möchte.

Das Umschalten der Antennen kann manuell mit einem kleinen Softwareprogramm erfolgen, es kann aber auch automatisch von der HackRF-One-Firmware auf Basis der Frequenz oder der Zeit durchgeführt werden. Frequenzbasiertes Umschalten ermöglicht zum Beispiel Breitband-Spektralanalyse-Anwendungen, während zeitbasiertes Umschalten solche coolen Dinge wie Pseudo-Doppler-Peilung ermöglicht [4].

Wenn Sie den Schalter im DP4T-Modus verwenden, können Sie ihn beispielsweise zum Einfügen von Dämpfungsgliedern oder Filtern in den Antennenpfad nutzen. Auf diese Weise kann der Opera Cake wie eine geschaltete Filterbank funktionieren.

## ... stapelbar

Außerdem können bis zu acht Opera Cakes übereinander gestapelt werden (yeah, ein geschichteter SDR-Kuchen...), was verschiedene Schaltkonfigurationen ermöglicht. Im manuellen Modus kann jede Platine separat gesteuert werden, aber im automatischen Frequenz- und Zeitschaltmodus schalten alle Karten zur gleichen Zeit und auf die gleiche Weise. Ein Stapel von zwei Platinen würde eine DP8T-Konfiguration ermöglichen, bei der zum Beispiel acht Filter geschaltet werden.

## Warum sollte ich ein Opera Cake haben wollen?

Opera Cake ist kein neuer Entwurf, er stammt aus dem Jahr 2016 und ist vielleicht sogar noch ein wenig älter. Neu ist, dass man ihn nicht mehr aus den auf GitHub veröffentlichten Designdateien zusammenbacken muss [5], sondern ihn jetzt als fix und fertiges, vollständig montiertes und getestetes Modul kaufen kann.

Der schön aufgebaute Opera-Cake Antennenschalter ist, wenig überraschend, ein perfekter Begleiter für das HackRF-One-SDR. Anstatt

die Hardware neu verdrahten oder die Länge der Antenne jedes Mal anpassen zu müssen, wenn Sie in einem anderen Band arbeiten wollen, können Sie nun bis zu acht „Lieblingsantennen“ auf einmal anschließen (und angeschlossen lassen).

### Ein paar Anmerkungen

Je nach Alter Ihres HackRF One müssen Sie möglicherweise die Firmware auf die neueste Version aktualisieren, damit er mit dem Antennenschalter funktioniert. Eine Anleitung dazu finden Sie auf der Website von Great Scott Gadgets [6].

Bitte beachten Sie, dass die SMA-Anschlüsse des Opera Cake, wie beim HackRF One, weiblich sind. Das bedeutet, dass er nur mit Antennen funktioniert, die mit einem männlichen Stecker ausgestattet sind, was man an dem Mittelstift in **Bild 2** leidlich gut erkennen kann. Seien Sie also vorsichtig und wählen Sie mit Bedacht!

Ein HackRF One mit einer oder mehreren darauf gestapelten Opera-Cake-Platinen passt nicht mehr in sein (enges) Gehäuse. Das macht das System natürlich anfällig für Staub, Schraubendreher und andere Metallgegenstände, die versehentlich hinein- und darauf fallen können. Je nachdem, wie Sie Ihre Antennen und Filter verdrahten, könnten auch die Reset- und Power-Tasten etwas unzugänglich werden (**Bild 3**). Da die stapelbaren Stiftleisten P20, P22 und P28 nur Strom und einige digitale Signale, aber keine HF-Signale übertragen, ist es möglich, zusätzlich Stiftleisten einzufügen, um den Abstand zwischen den Platinen zu vergrößern. Ein (kurzes) männlich-männliches SMA-Kabel zum Anschluss des Opera Cake an den Antenneneingang des HackRF One ist leider nicht im Lieferumfang enthalten. ◀

(220602-02)RG

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Dann kontaktieren Sie Clemens Valens unter [clemens.valens@elektor.com](mailto:clemens.valens@elektor.com)!



Bild 1. Die Opera-Cake-Platine von Great Scott Gadgets ist ein Antennenschalter für HackRF One.



Bild 2. Antennen mit SMA-Anschlüssen. Links ist der richtige und rechts der falsche Anschluss für Opera Cake und HackRF One zu sehen.

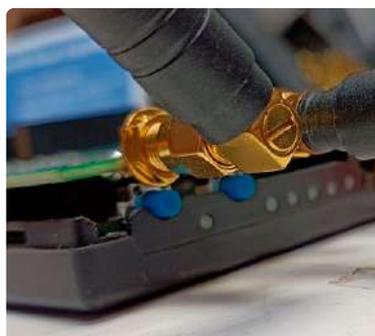


Bild 3. Um den Abstand zwischen den beiden Platinen zu vergrößern, können Sie zusätzlich stapelbare Header einfügen.



### Passende Produkte

- Opera Cake - Antennenschalter für HackRF One (SKU 20083) [www.elektor.de/20083](http://www.elektor.de/20083)
- HackRF One Software Defined Radio (1 MHz bis 6 GHz) (SKU 18306) [www.elektor.de/18306](http://www.elektor.de/18306)
- ANT500 Teleskopantenne (75 MHz bis 1 GHz) (SKU 18481) [www.elektor.de/18481](http://www.elektor.de/18481)
- ANT700 Teleskopantenne (300 MHz bis 1100 MHz) (SKU 18480) [www.elektor.de/18480](http://www.elektor.de/18480)



### WEBLINKS

- [1] Snickerdoodle auf Elektor Labs: <https://www.elektormagazine.de/labs/snickerdoodles-with-zynq>
- [2] D. Meyer, „HackRF One SDR Transceiver: From 1 MHz to 6 GHz“, Elektormagazin.com: <https://www.elektormagazine.com/news/hack-rf-one-sdr-transceiver>
- [3] C. Valens, „Introducing the GreatFET One Dual-Core Microcontroller Board for Python“, Elektormagazine.com: <https://www.elektormagazine.com/news/greatfet-one>
- [4] Pseudo-Doppler-Peilung: <https://www.rtl-sdr.com/pseudo-doppler-direction-finding-with-a-hackrf-and-opera-cake/>
- [5] Opera Cake auf GitHub: <https://github.com/greatscottgadgets/hackrf/tree/master/hardware/operacake>
- [6] Firmware für HackRF One: <https://github.com/greatscottgadgets/hackrf/releases/latest>

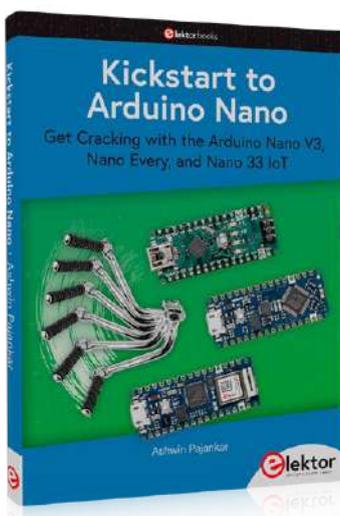


# Technik mit **Arduino** und mehr

Ein Interview mit Autor Ashwin Pajankar

Von Alina Neacsu (Elektor)

Die Elektor-Community ist voll von Ingenieuren, die ihr Wissen gerne mit gleichgesinnten Elektronikfans teilen. So wie Ashwin Pajankar, ein Ingenieur, Pädagoge, Elektor-Autor und YouTuber aus Nashik in Indien. Wenn er nicht gerade an seinem Labortisch ein neues Elektronikprojekt in Angriff nimmt, hilft er Gleichgesinnten mit Büchern, Kursen und Videos, den Umgang mit Arduino und anderen innovativen Technologien zu erlernen.



Lesen Sie *Kickstart to Arduino Nano* und beginnen Sie Ihre Reise mit dem *Arduino Nano V3, Nano Every* und *Nano 33 IoT*.

**Alina Neacsu:** Zunächst einmal vielen Dank, dass du dir die Zeit nimmst, diese Fragen zu beantworten. Erzählst du uns etwas über dich? Was ist dein derzeitiger Beruf?

**Ashwin Pajankar:** Ich lebe in den Außenbezirken der Stadt Nashik in Indien. Ich habe einen Bachelor of Engineering in Informatik und Ingenieurwesen beim Shri Guru Gobind Singhji Institute of Engineering and Technology in Nanded erworben. Außerdem halte ich einen Master of Technology (ebenfalls in Informatik und Ingenieurwesen) am International Institute of Information Technology in Hyderabad. Ich spreche fünf Sprachen und habe in drei Bundesstaaten Indiens gelebt und gearbeitet.

Derzeit arbeite ich als freiberuflicher technischer Autor, YouTuber und Dozent bei Udemy. Außerdem organisiere ich Bootcamps für Programmierung und Elektronik für Berufstätige und Studenten. Und ich liebe Mathe, Physik, Computer und Elektronik!

**Alina:** Kannst du dich an dein erstes Elektronikprojekt mit Mikrocontrollern erinnern? Kannst du es uns beschreiben und sagen, welche Technologien verwendet wurden?

**Ashwin:** Mein allererstes Mikrocontroller-Projekt war sehr bescheiden. Ich habe in der 12. Klasse mit 8051-Mikrocontrollern angefangen (zusammen 8085 und 8086), was damals der Standard-Mikrocontroller war, der in ganz Indien für den Unterricht verwendet wurde. Bis zur 12. Klasse hatten wir jedoch keine praktischen Erfahrungen damit. Als ich Ingenieurwesen studierte, ein vierjähriges Bachelor-Studium, hatte ich im dritten Studienjahr die Möglichkeit, mit dem 8051-Kit in Assembler zu arbeiten. Mein allererstes Projekt mit dem 8051 war ein sehr einfaches - das Blinken einer LED mit 8051-Assemblercode.

**Alina:** Das Arduino-Ökosystem – kannst du uns sagen, wie es funktioniert? Und was sind die Vor- und Nachteile?

**Ashwin:** Arduino ist ein Open-Source-Ökosystem für Hardware und Software. Da das Design quelloffen ist, können Menschen und Organisationen ihre eigenen Boards herstellen und Ideen mit der Community teilen. Die Software ist ebenfalls quelloffen, und der Quellcode ist unter [github.com/arduino](https://github.com/arduino) zu finden. Man kann zu diesem Hauptzweig beitragen oder das Projekt abspalten, um eine eigene Variante zu entwickeln. Im Internet gibt es eine Unzahl von Lernressourcen. Mit Arduino kann man gleichzeitig in die Elektronik und die C-Programmierung einsteigen, ohne astronomische Summen



ausgeben zu müssen. Dies sind die Vorteile des Arduino-Ökosystems. Der einzige Nachteil, der mir einfällt, ist, dass manchmal gefälschte Produkte, die als originale Arduino-Produkte gekennzeichnet sind, an ahnungslose Enthusiasten verkauft werden.

**Alina:** Was hat dein Interesse am Board Arduino Nano 33 IoT geweckt?

**Ashwin:** Nun, wenn wir früher ein Arduino-Board mit einem WLAN verbinden wollten, mussten wir spezielle Shields verwenden, die in Indien nur schwer zu bekommen sind, oder das ESP-01-Board benutzen. Die Arbeit mit dem ESP-01 ist etwas komplizierter, und Anfänger sind damit oft überfordert. Der Arduino Nano 33 IoT dagegen ist eine perfekte IoT-Lösung und sofort einsatzbereit. In ihm steckt ein WLAN-Modul NINA-W102. Ich schaue regelmäßig auf [arduino.cc](https://www.arduino.cc) nach neuen Produkten und war sehr aufgeregt, als ich das neue Board mit eingebautem WLAN entdeckte. Als es dann in Indien verfügbar wurde, habe ich es bei der ersten Gelegenheit gekauft und damit experimentiert.

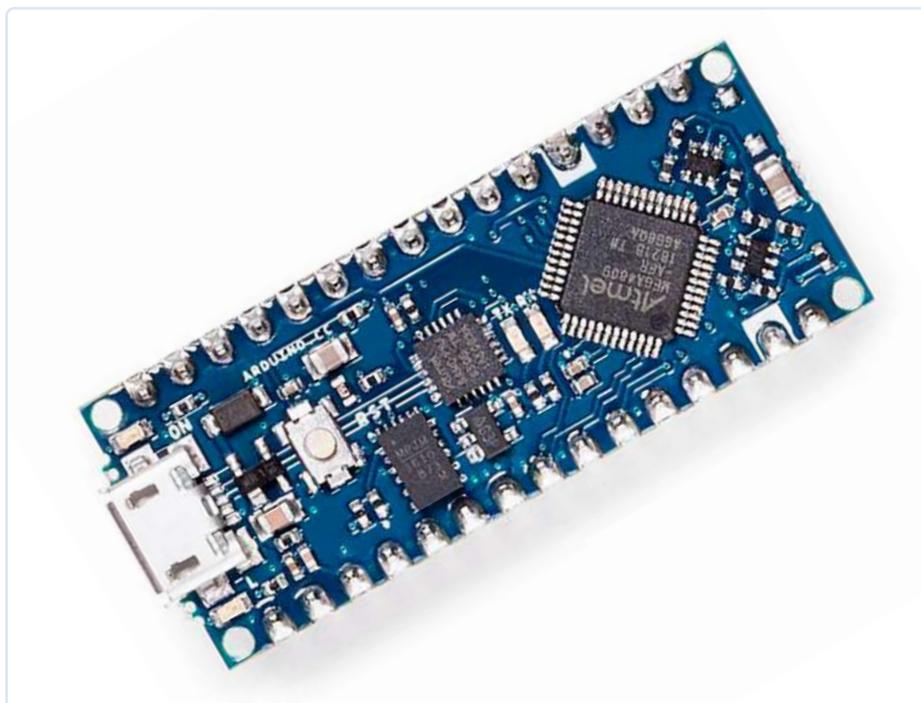
**Alina:** Hast du einen Rat für alle, die mit dem Arduino Nano anfangen wollen?

**Ashwin:** Ja. Sehen Sie sich die Online-Dokumentation auf der Arduino-Homepage an. Mein Buch *Kickstart to Arduino Nano* (Elektor 2022) enthält eine detaillierte Schritt-für-Schritt-Anleitung.

**Alina:** Erzähle uns etwas darüber, wie du dein erstes Buch geschrieben hast. War es eine Herausforderung?

**Ashwin:** Ich arbeitete als IT-Ingenieur im indischen Bangalore, als ich mein allererstes Buch schrieb und veröffentlichte. Das Buch handelte vom Raspberry Pi und befasste sich mit Computer Vision. Obwohl es meine erste Erfahrung mit der Veröffentlichung eines Buches war, empfand ich es nicht als große Herausforderung, da ich das Programmieren, die Elektronik und das Schreiben liebe. Das Verfassen von Büchern ermöglicht es mir, all meine Leidenschaften zu etwas sehr Greifbarem zu verbinden.

**Alina:** Du erwähnst in deinem Buch, dass du während des Schreibens in Indien warst,



*Ashwin Pandakar bietet in seinem Buch einen umfassenden Einstieg in den Arduino Nano.*

wo du auch zurzeit lebst. Wie kam das Buch während der COVID-19-Pandemie zustande?

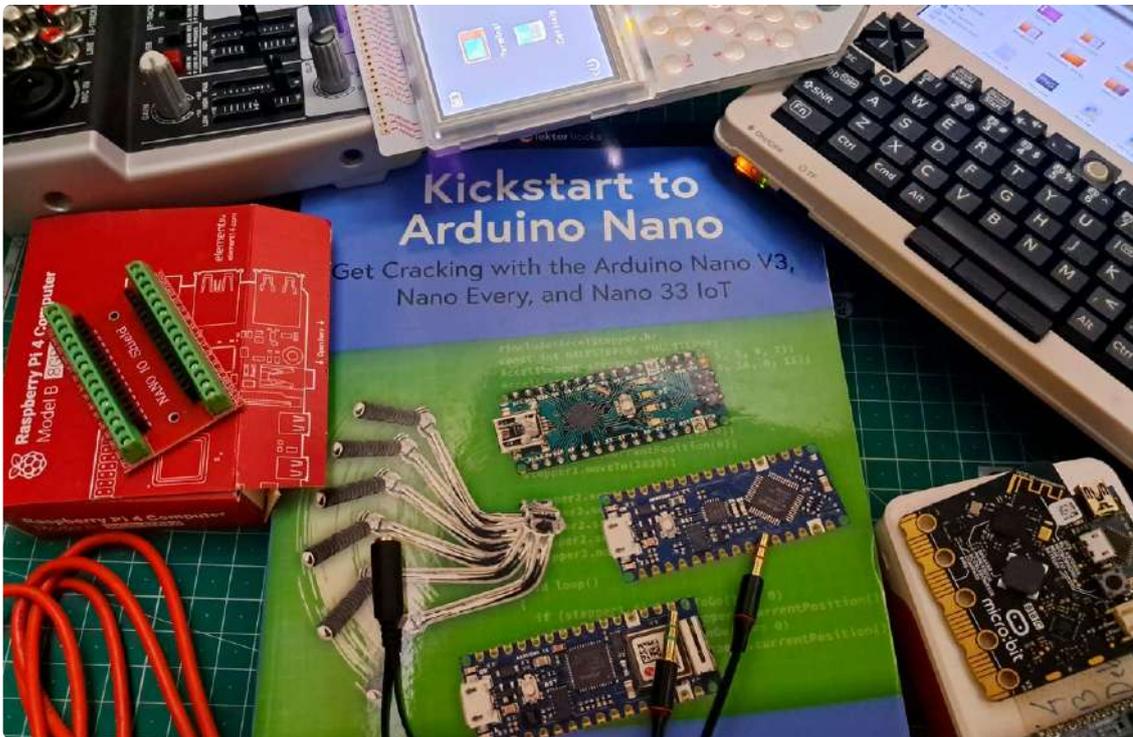
**Ashwin:** Die COVID-19-Pandemie war eine schwere Zeit für unsere gesamte menschliche Zivilisation. Indien war eines der am schlimmsten betroffenen Länder, mit einer der höchsten Hospitalisierungsraten und einer der höchsten Sterbeziffern. Und Nashik, die Stadt, in der ich derzeit wohne, hatte die höchste Todesrate in Indien. Es war sehr beunruhigend, denn niemand hatte je zuvor eine solche Katastrophe erlebt. Es gab ständige Abriegelungen, Ausgangssperren und Einschränkungen. Es fehlte ständig an Lebensmitteln, Medikamenten und anderen Hilfsgütern. In den Krankenhäusern waren keine Betten verfügbar, Krankenwagen und Leichenwagen waren überlastet. Auch ich erkrankte zweimal an der Krankheit. Beim ersten Mal wurde ich in ein staatliches Krankenhaus eingeliefert, beim zweiten Mal wurde ich zu Hause behandelt. Ich erinnere mich noch genau daran, wie ich herumliefe, um Grippemittel für mich und meine Nachbarn zu besorgen, da die Medikamente sehr knapp waren.

Das Schreiben des Buches gab mir in diesen dunklen Zeiten einen Lebensinhalt. Durch

die Isolation und den Beinahe-Stillstand fast aller anderen Aspekte des Lebens hatte ich die Möglichkeit, mich auf das Schreiben des Buches zu konzentrieren. Außerdem war das Team von Elektor sehr hilfreich und hat mich bei jedem Schritt begleitet. Ehrlich gesagt fehlen mir die Worte, um meine tiefe Dankbarkeit für die großartige Unterstützung auszudrücken, die sie mir immer gegeben haben.

**Alina:** Du engagierst dich sehr für den Online-Unterricht, insbesondere über die Online-Kurse bei Udemy. Kannst du uns diese Art von Lehren und Lernen beschreiben? Wie kommen die Studenten mit dir in Kontakt, verglichen mit einem traditionellen Kurs?

**Ashwin:** Ich bevorzuge es, online zu unterrichten, da ich so eine größere Reichweite habe. Ich habe in der Vergangenheit auch Programmier-Bootcamps für Live-Publikum durchgeführt. Mit Online-Unterricht erreiche ich aber Hunderttausende von Studenten. Die Leute können mit mir in Kontakt treten, indem sie mir auf dem Portal Fragen stellen. Und da alle Kurse zum Selbststudium sind, ist das Engagement der Studierenden viel höher.



Der Labortisch von Ashwin Pajankar mit verschiedenen Geräten.

**Alina:** Wann hast du gemerkt, dass es dir Spaß macht, andere Menschen über deine Lieblingsthemen zu unterrichten?

**Ashwin:** Während der Sommerferien meines Ingenieurstudiums unterrichtete ich die Schüler der Oberstufe und half ihnen bei der Vorbereitung auf die Aufnahmeprüfungen für Ingenieure. Ich habe Mathe, Physik, Informatik und Englisch unterrichtet und dabei gemerkt, dass sehr gerne unterrichte und Wissen vermittele.

**Alina:** Was ist letztendlich das Wichtigste, das du den Lesern deines Buches mit auf den Weg geben möchtest?

**Ashwin:** Das Wichtigste, was ich den Lesern mit auf den Weg geben möchte, ist, dass sie lernen, die Welt der Technologie selbst zu erkunden. Die Welt der Technologie ist sehr leicht zu durchschauen, wenn man weiß, wo man nach relevantem Wissen suchen muss. Um mehr zu lernen, sollten die Leser die gesamte Online-Dokumentation, Codebeispiele und verschiedene technische Diskussionsforen selbst erkunden.

**Alina:** Arbeitest du derzeit an etwas Speziellem? Irgendwelche neuen Ideen für ein Buch?

**Ashwin:** Ja. Ich habe mit dem Raspberry Pi 4 gearbeitet. Während des Lockdowns war er sehr schwierig zu bekommen. In letzter Zeit hat sich die Situation jedoch verbessert, und ich konnte einen Raspberry Pi 4 mit 8 GB RAM beschaffen. Ich habe mit ihm experimentiert. Sobald ich das

persönliche Projekt, an dem ich arbeite, abgeschlossen habe, werde ich ein Buch über Raspberry Pi für Elektor schreiben können. Ich liebe das Buchformat von Elektor, und Elektor hat das beste Team. Ich habe vor, mit Elektor noch viele weitere technische Bücher zu verschiedenen Themen zu veröffentlichen.

**Alina:** Kannst du uns etwas über persönliche Projekte erzählen, an denen du gerade arbeitest?

**Ashwin:** Zurzeit habe ich mich auf die große Reise begeben, meinen YouTube-Kanal zu erneuern und sein Publikum zu vergrößern. Ich werde bald viele Videos hochladen, in denen ich Mathe auf Schulniveau in Hindi unterrichte, einer Sprache, die von einer Milliarde Menschen auf der ganzen Welt gesprochen wird. Danach werde ich mich mit den Grundlagen der Physik und der Elektrotechnik beschäftigen, ebenfalls in Hindi.

**Alina:** Gibt es eine Leistung oder einen Beitrag, auf den du besonders stolz bist?

**Ashwin:** Mit Mathematik, Programmierung und Elektronik verdiene ich meinen Lebensunterhalt. Abgesehen von meiner beruflichen Arbeit habe ich mich aber auch immer aktiv für die Gesellschaft engagiert, indem ich mich an meinen Arbeitsplätzen und an den Universitäten, an denen ich studiert habe, für soziale Belange eingesetzt habe. Seit ich als Freiberufler arbeite, tue ich dies auch in meinem persönlichen Umfeld. Aufgrund meiner Teilnahme an

einem Bildungsprogramm für unterprivilegierte Kinder in der Gemeinde im Rahmen des Community-Outreach-Programms meiner Universität wurde mein Interview in einer der bekanntesten Zeitungen von Hyderabad veröffentlicht. Ich habe mehrere Auszeichnungen für mein gesellschaftliches Engagement erhalten. Das sind die Dinge in meinem Leben, auf die ich am meisten stolz bin. ◀

(220626-02)RG

### Haben Sie Fragen oder Kommentare?

Wenn Sie Fragen zu diesem Interview haben, senden Sie bitte eine E-Mail an Alina Neacsu unter [alina.neacsu@elektor.com](mailto:alina.neacsu@elektor.com) oder an das Elektor-Redaktionsteam unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



### Passende Produkte

- **A. Pajankar, Kickstart to Arduino Nano (Elektor 2022)**  
Buch (Paperback), SKU20241:  
[www.elektor.de/20241](http://www.elektor.de/20241)  
E-Buch (PDF), SKU 20242:  
[www.elektor.de/20242](http://www.elektor.de/20242)
- **Arduino Nano Every mit Header (SKU 19939)**  
[www.elektor.de/19939](http://www.elektor.de/19939)

# LiDAR-Präzisionsmetermaß

Misst bis zu zwölf Meter



Von Somnath Bera (Indien)

LiDAR-Module sind großartig zum Erkennen von Hindernissen und zum Messen von Entfernungen. Hier verwenden wir ein LiDAR-Modul, um präzise Entfernungsmessungen bis zu 12 m mit einer Auflösung von 1 cm vorzunehmen.

Ursprünglich habe ich ein Benewake-LiDAR mit der Bezeichnung TFMMini-S gekauft, um damit eine Art Radar zu bauen. Aber wie so oft wurde aus den Plänen nichts, bis ich bei meiner Arbeit eine aufschlussreiche Erfahrung machte und ich das Modul anderweitig nutzen konnte. Elektrisch widerstandsgeschweißte Rohre (ERW) sind lange Rohre mit einem Durchmesser von 300 mm oder mehr und einer Länge von bis zu zwölf Metern. Sie werden in unserem Kraftwerk für den Transport von Ascheschlamm verwendet. Eines Tages, als ich mit diesen langen Rohren für die Montage der Ascheabzugsleitungen hantierte, stellte ich fest, dass mein Team jedes einzelne Rohr genau vermessen musste, bevor es durch das Ausgangstor herausgeschafft wurde. Jeden Tag musste sich jemand mindestens 30...50 Mal die Zeit nehmen, diese Messungen vorzunehmen. Unter freiem Himmel und bei 47°C in der sengenden Sonne das Maßband an einem Ende zu halten und am anderen Ende eine Aufsichtsperson, die dann die Länge aufzeichnet, das ist eine sehr mühsame Arbeit. Hier entstand die Idee, ein Präzisionsmessgerät zu entwickeln.

## LiDAR-Module

Das Single-Point-Ranging-LiDAR TFMMini-S (**Bild 1**) kann Entfernungen bis zu 12 m sehr schnell (100 Hz) und sehr genau (1 cm Auflösung, 1 % Fehler) messen, was für diese Aufgabe genau richtig ist. Es gibt auch eine Plus-Version, die schneller ist (1 kHz) und in einem IP65-geschützten Gehäuse steckt. Beide Module sind mit einer seriellen Schnittstelle ausgestattet.

Wenn man das LiDAR an einem Ende des Rohrs anbringt und es dann auf eine gewinkelte Platte am anderen Ende richtet, erhält man die genaue Länge des Rohrs. Beide Versionen des LiDAR-Moduls haben ein Sichtfeld (FOV) von 3,4 Grad, so dass die korrekte Ausrichtung sehr wichtig ist, um ebenso korrekte Messwerte zu erhalten. Um dies

zu erleichtern, haben wir unser System mit einem Laserpointer ausgestattet, der mit einem Drucktaster aktiviert wird.

Ich habe ein ESP32-Modul zur Steuerung des LiDARs und zur Durchführung der Berechnungen verwendet. Ein kleines OLED-Display wurde hinzugefügt, um die Entfernungswerte anzuzeigen. Die Einbeziehung eines gleitenden Mittelwerts einiger Messungen erhöht die Genauigkeit. Das Ergebnis könnte dann sogar als Fließkommawert ausgegeben werden, aber der Schnelligkeit und Einfachheit der Messungen wegen habe ich es bei reinen Ganzzahlwerten belassen.

Die TFMMini-LiDARs nutzen die Reflexion von unsichtbarem Infrarot-Laserlicht an der Oberfläche des Objekts, auf das sie gerichtet sind. Wenn das Objekt das einfallende Laserlicht nicht reflektiert oder vollständig absorbiert, ist die Messung fehlerhaft. Vermeiden Sie zum Beispiel Wasser und schräge, stark reflektierende Glasfenster. Die meisten anderen Objekte reflektieren genügend Licht, so dass die Messungen problemlos durchgeführt werden können, auch wenn sich das Objekt bewegt.



*Bild 1. Das kostengünstige LiDAR-Modul TFMMini-S hat eine Reichweite von 12 m und eine Auflösung von 1 cm.*

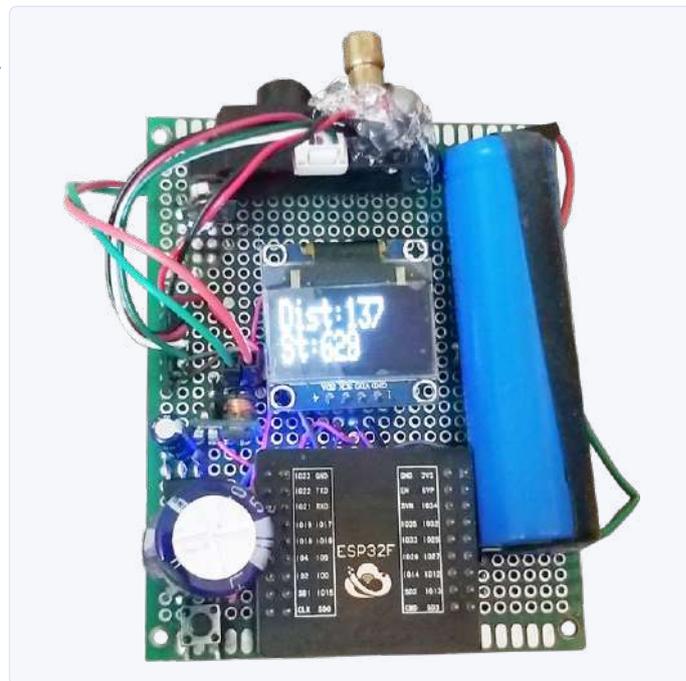
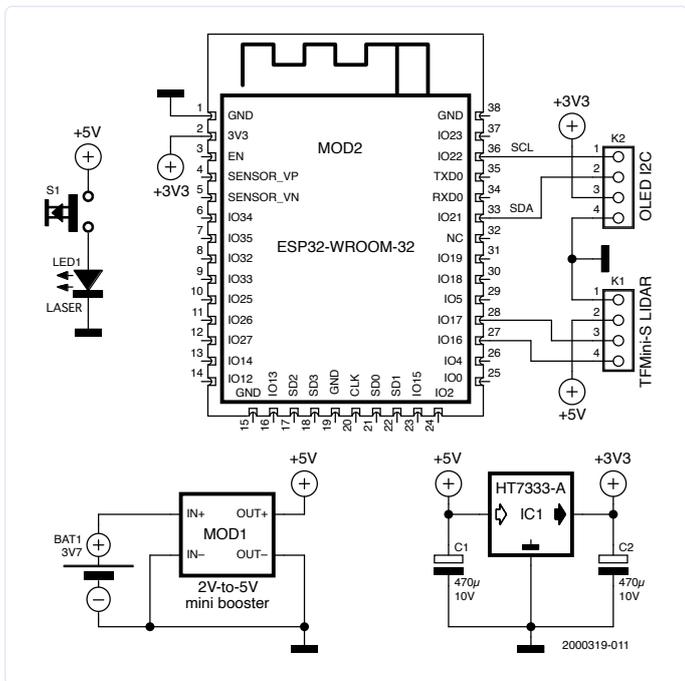


Bild 2. Der Prototyp verwendet ein ESP32-WROOM-Modul ohne integrierten 5-V-Regler. Bei einem DevMod-C- oder Pico-Kit-Modul kann IC1 weggelassen werden. In diesem Fall schließen Sie die 5-V-Versorgung an den 5-V-Eingang des Moduls an, nicht an dessen 3,3-V-Pin!

Bild 3. Der Prototyp des LiDAR-basierten Präzisionsmessgeräts wurde auf einer gewöhnlichen Lochraster-Platine aufgebaut. Beachten Sie, wie der Laserpointer auf das LiDAR-Modul aufgeklebt ist.

### Bau des Entfernungsmessers

Beim Bau des Geräts musste ich bedenken, dass es portabel sein muss, damit mein Team es im Feld einsetzen kann, also batteriebetrieben sein. Die Schaltung, die ich schließlich entwickelt habe, ist in **Bild 2** zu sehen. Sowohl das LiDAR als auch die Laserdiode, die ich zum Ausrichten verwendet habe, benötigen eine Versorgungsspannung von 5 V, das ESP32-WROOM-Modul aber nur 3,3 V. Daher habe ich einen Mini-Aufwärtswandler hinzugefügt, der aus einer Eingangsspannung von nur 2 V eine 5-V-Versorgung erzeugen kann. An den 5-V-Ausgang habe ich dann einen Low-Dropout-Spannungsregler HT7333-A angeschlossen, der die 3,3 V für den ESP32 erzeugt. Wenn Sie ein ESP32-Modul mit integriertem Spannungsregler verwenden, dann brauchen Sie den HT7333-A natürlich nicht. Der komplette

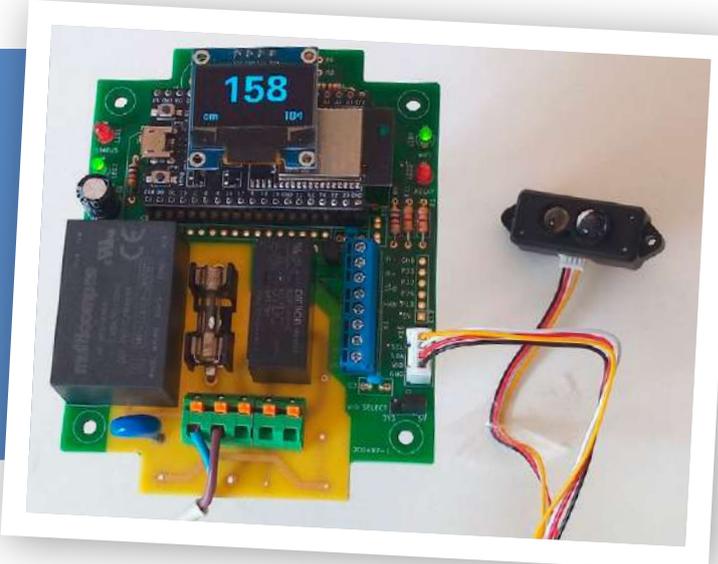
Aufbau läuft mit zwei 1,5-Volt-Batterien oder mit einem 3,7-Volt-LiPo-Akku. Beachten Sie, dass das HT7333-A ein SMD ist; seien Sie daher vorsichtig bei der Bestückung!

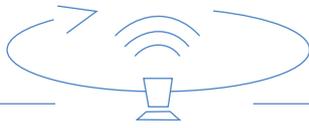
### Ein Wort zur Software

Die Software, die ich für den LiDAR-Zollstock geschrieben habe, ist ein Arduino-Sketch und kann von [1] heruntergeladen werden. Sie benötigt die TFMMini-Bibliothek und die GFX SSD1306-Bibliotheken von Adafruit, die über den Bibliotheksmanager der Arduino-IDE verfügbar sind. Das LiDAR und der ESP32 kommunizieren über eine serielle Schnittstelle (Serial1) miteinander. Alle 25 ms wird ein neuer Datenpunkt angefordert. Die ganze harte Arbeit wird von der TFMMini-Bibliothek erledigt, während der Sketch lediglich eine Benutzeroberfläche bietet.

### Nutzen Sie den Automator!

Der Elektor-Automator [2] ist die perfekte Plattform, um mit dem TFMMini-S-LiDAR zu spielen. Er basiert auf einem ESP32-Modul, hat einen Grove-Anschluss, der mit dem LiDAR-Modul kompatibel ist, und verfügt über ein OLED-Display und die notwendigen Stromversorgungseinrichtungen. Im Elektor-Labor haben daher einen Arduino-Sketch für dieses Projekt erstellt, das auf dem Automator läuft. Es verwendet die Bibliotheken *TFLidar* und *U8g2*, die über den Bibliotheksmanager der Arduino-IDE verfügbar sind. Die Software kann leicht erweitert werden, um WLAN- oder Bluetooth-Konnektivität hinzuzufügen und um das Relais und die LEDs des Automators zu steuern. Sie kann unter [2] heruntergeladen werden.





Entfernungs- und Signalstärkewerte werden auf dem OLED-Bildschirm angezeigt und auch über die serielle Schnittstelle des Arduino (Serial0) übertragen.

### Verwendung des Messgeräts

Nachdem Sie alle Teile auf einer Lochrasterplatine aufgebaut haben (**Bild 3**), positionieren Sie den LiDAR so am Rand der Platine, dass er einen freien Blick auf das Objekt hat. Der Laserpointer ist eine einfache rote Laserdiode, die durch einen Drucktaster aktiviert wird. Wenn möglich, befestigen Sie sie mit Heißkleber (oder ähnlichem) so auf dem LiDAR, dass beide Lichtquellen in die gleiche Richtung zeigen. Die Mindestentfernung, die das LiDAR-Modul TFMini-S bewältigen kann, ist 30 cm, so dass der Messbereich 30 cm bis 12 m beträgt. Messen Sie die Entfernung und überprüfen Sie sie mit einem guten Maßband. Mit der Zeit werden Sie feststellen, dass die Messungen bis auf den Zentimeter stimmen!

Die zweite Zeile auf dem Display zeigt die Stärke des empfangenen Signals an. Wenn sich das Objekt entfernt, nimmt die Stärke ab. ◀

200316-02

### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Kontaktieren Sie Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).



### PASSENDE PRODUKTE

- > **TFMini-S LiDAR (SKU 19691)**  
[www.elektor.de/19691](http://www.elektor.de/19691)
- > **TFmini Plus IP65 LiDAR (SKU 19690)**  
[www.elektor.de/19690](http://www.elektor.de/19690)
- > **ESP32-WROOM-32 (SKU 18421)**  
[www.elektor.de/18421](http://www.elektor.de/18421)
- > **0.96" OLED Display (blau, I<sup>2</sup>C, 4-Pin) (SKU 18747)**  
[www.elektor.de/18747](http://www.elektor.de/18747)

### WEBLINKS

- [1] Dieses Projekt bei Elektor Labs: [www.elektormagazine.de/labs/tfmini-lidar-precision-gauge-30-cm-to-1200-cm](http://www.elektormagazine.de/labs/tfmini-lidar-precision-gauge-30-cm-to-1200-cm)
- [2] Elektor-Automator: [www.elektormagazine.de/labs/automator](http://www.elektormagazine.de/labs/automator)

# Bestückung für Kleinserien und Prototypen



## M10V & MX70

Die vollautomatischen,  
preiswerten Baugruppenbestücker

- SMD-Bestücker für Kleinserien und Prototypen
- mit optionalem Präzisions-Dispenser für Paste und Kleber
- Spektrum: 0201 ... > 35 x 35 mm
- optische Zentrierung der Bauteile
- Vision-System zur Entnahme aus Schüttgut-Behältern
- Ideal auch für Schulen und Ausbildungsstätten



**factronix**<sup>®</sup>  
Systeme für die Elektronikfertigung

Tel. +49 (0)8153/90 664-0  
[office@factronix.com](mailto:office@factronix.com)  
[www.factronix.com](http://www.factronix.com)

# Audio mit dem ESP32

## Das Framework ESP-ADF in der Praxis

Von Tam Hanna (Ungarn)

Consumerelektronik ist ein herausfordernder Markt: Neben extrem schnellen Produktzyklen gilt insbesondere im Audibereich, dass die Ansprüche an die verwendeten Algorithmen (Stichwort Codecs) nicht ohne sind. Mit ESP-ADF stellt Espressif ein leistungsfähiges Paket zur Verfügung, das Entwicklern Zeit und Aufwand spart.

Das *Audio Development Framework* von Espressif ist eine Sammlung von Algorithmen und Codecs, die in einem standardisierten Format vorliegen. Der Entwickler realisiert seine Aufgabe durch das „In-Serie-Schalten von Software-ICs“, ohne sich mit Details auseinandersetzen zu müssen.

Ziel dieses Artikels ist die Vermittlung von Grundverständnis darüber, wie die Arbeit mit dem ADF in der Praxis aussieht. Der Autor dieser Zeilen nutzt das Framework ADF selbst intensiv in einem Projekt für einen Beratungskunden – und präsentiert hier sozusagen eine Liste von Allem, was er gerne von vornherein gewusst hätte.

### Von der Hardware

Beginnen wir mit dem Offensichtlichen: ESP-ADF ist mit jedem ESP32-basierenden System gleichermaßen kompatibel. Für den „Audioteil“ bietet die Bibliothek eine standardisierte Treiberoberfläche an, die die Bereitstellung der Daten übernimmt. Die praktische Erfahrung lehrt, dass zumindest Teile der Software auch ohne Vorliegen der spezifischen finalen Hardware entwickelt werden können. Hierfür bietet Espressif spezielle Entwicklerboards an. Der Autor verwendet in seinem Unternehmen gern das in **Bild 1** und **Bild 2** gezeigte LyraT.

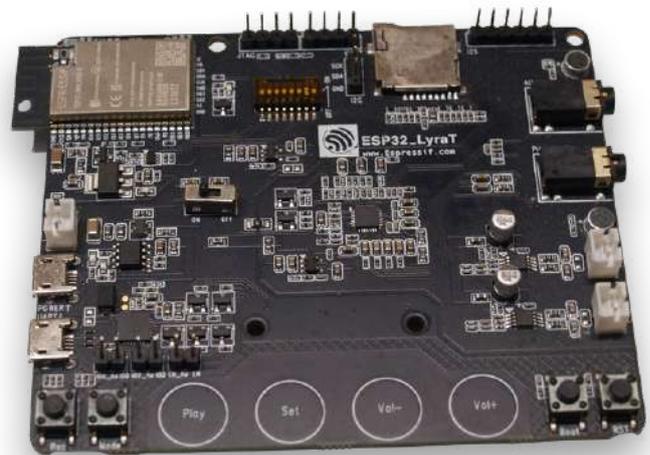


Bild 1: Die kleine schwarze Platine bringt alles mit...

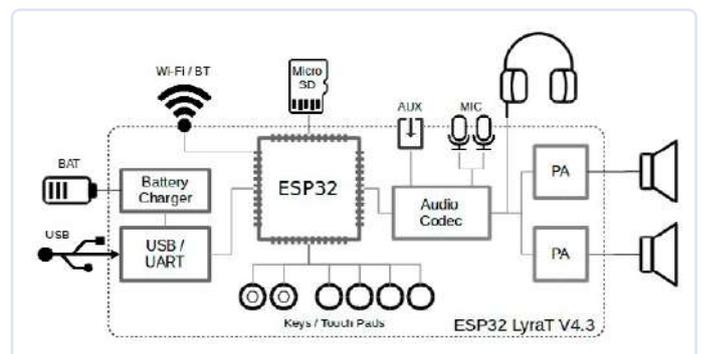


Bild 2: ...was man zum Prototyping von Audioapplikationen benötigt (Quelle: [6]).

Am wichtigsten ist das „Analog Baseband“, das in Form einer auf dem ES8388 vorliegenden Schaltung (neben zwei analogen Mikrofonen) vorliegt. Die Kommunikation erfolgt dabei sowohl über I2S als auch durch eine Gruppe dedizierter Pins – das Schaltbild wird von Espressif zur Verfügung gestellt und lässt sich problemlos nachbauen. Der Chip ist bei UTSOURCE und LCSC ohne große Probleme erhältlich.



„Kritisch“ ist bei der Arbeit mit dem LyraT vor allem das am Ende doch limitierte GPIO-Komplement des ESP32. Das Anschließen von größeren Mengen an externer Peripherie scheitert mitunter daran – notfalls bietet sich die Nutzung eines Zweitprozessors an, der sich um die Hardwarekommunikation kümmert.

## Einrichtung des ADF

Espressif stellt das ADF als Plug-In für die bekannte IDF-Umgebung zur Verfügung: wer mit dem (nach Ansicht des Autors nicht empfehlenswerten) Arduino-Core arbeitet, hat Pech gehabt. Problematisch ist in diesem Zusammenhang auch, dass nicht jede Version von IDF als „Host“ für ADF tauglich ist.

Der bequemste Weg, um eine lauffähige Arbeitsumgebung zu erhalten, ist das Herunterladen eines schlüsselfertigen Pakets. Der Autor legt es in den folgenden Schritten im Unterordner `esp4` ab:

```
(base) tamhan@tamhan-thinkpad:~$ mkdir esp4
(base) tamhan@tamhan-thinkpad:~$ cd esp4/
```

Der eigentliche Download erfolgt dann unter Nutzung des Git-Kommandozeilenclients:

```
(base) tamhan@tamhan-thinkpad:~/esp4$ git clone
--recursive https://github.com/espressif/esp-adf.git
Cloning into 'esp-adf'...
```

Sorgfältige Betrachtung der (hier nicht abgedruckten) Ausgabe zeigt, dass das ADF-Repositorium diverse Verweise aufweist – aus diesem Grund ist ein Download über die in GitHub integrierte Browserdownloadfunktion nicht möglich.

Dem ADF liegt eine IDF-Version bei, einige der Komponenten verlangen das Vorhandensein der Umgebungsvariable `ADF_PATH`. Sofern Sie ein Kommandozeilenfenster für die Verarbeitung von ADF-basierten Applikationen verwenden wollen, müssen Sie die Umgebungsvariable nach folgendem Schema setzen:

```
(base) tamhan@tamhan-thinkpad:~/esp4$ cd esp-adf
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf$ export
ADF_PATH=$PWD
```

Das Vorhandensein der Variable `ADF_PATH` wirkt sich manchmal auf normale IDF-Projekte aus: Der (grundparanoide) Autor achtet darauf, für die „normale“ ESP32-Arbeit ein anderes Terminalfenster zu verwenden.

Wegen der integrierten IDF-Version ist im nächsten Schritt der übliche Install-Lauf erforderlich, um Compiler und andere Abhängigkeiten bereitzustellen:

```
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf$ cd
$ADF_PATH/esp-idf
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/esp-idf$
./install.sh
```

```
Detecting the Python interpreter
. . .All done! You can now run:
```

```
./export.sh
```

Für die eigentliche Einrichtung der Toolchain ist dann noch ein Aufruf eines Skripts erforderlich – die beiden Punkte sind kein Druckfehler:

```
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/esp-idf$
. ./export.sh
```

In der praktischen Entwicklung ist es an dieser Stelle ratsam, ein Shellskript zur Parametrierung anzulegen – die Bash kann sich dann auf Zuruf selbsttätig konfigurieren.

## Softwarearchitektur: Pipeline

Nach dem Herunterladen und Einrichten der ADF ist es an der Zeit, Überlegungen zum theoretischen Aufbau des Frameworks anzustellen. Getreu dem einst bei *The Stepstone Corporation* entwickelten Begriff des „Software-ICs“ gilt auch im Fall des ADF, dass der Entwickler im Prinzip eine aus einer Gruppe von Verarbeitungsschritten bestehende Pipeline realisiert. Zur Veranschaulichung zeigt **Bild 3** einen Workflow, der einen MP3-Player implementiert. Im Bereich der eigentlichen Pipelineelemente wagt Espressif ebenfalls keine Experimente – **Bild 4** zeigt die angebotenen Rollen und Beispielimplementierungen.

Besonders ist am vorliegenden System eigentlich nur, dass der von Unix bekannte Ansatz des „alles ist eine Datei“ auch für Peripheriegeräte gilt. Die unter [1] en détail dokumentierte und anhand des unter [2] bereitstehenden Buttons am einfachsten verständliche API erlaubt das Erzeugen von „Wrappern“, die sich direkt in die Pipeline integrieren lassen. Für Anwendungsentwickler ist dies allerdings nur insofern relevant, als dass der Gedanke, Peripheriegeräte ebenfalls als Pipelineelemente anzusehen, mitunter etwas Denkarbeit voraussetzt.

Unter [3] finden sich API-Dokumentationen zu den einzelnen Elementen der Pipeline.

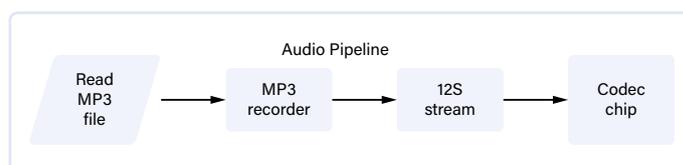


Bild 3: Diese Pipeline decodiert MP3s (Quelle: [3]).

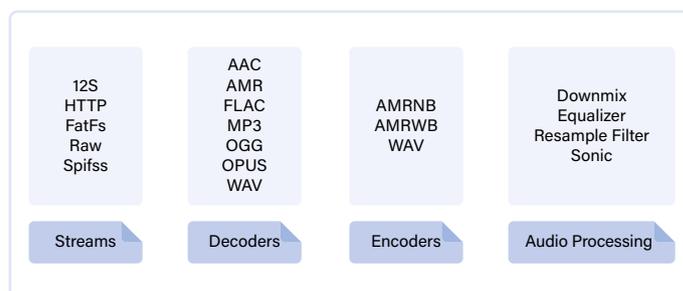


Bild 4: Pipelineelemente lassen sich in mehrere Typen unterteilen (Quelle: [3]).

## Softwarearchitektur: Praxisversuch MP3-Player

Zum praktischen Verständnis wollen wir uns eines der von Espressif bereitgestellten Beispiele ansehen: Wer ein auf dem ADF basierendes Projekt realisieren möchte, ist gut beraten, im ersten Schritt nach einer geeigneten oder zumindest artverwandten Vorlage zu suchen. Das Beispielverzeichnis lässt sich im Terminal durch Eingabe des folgenden Kommandos öffnen:

```
cd $ADF_PATH/examples/
```

Wir wollen in den folgenden Schritten den klassischen MP3-Player ansehen. Wechseln Sie in das Verzeichnis `~/esp4/esp-adf/examples/get-started/play_mp3_control/main`, und öffnen Sie die Datei `play_mp3_control_example.c` in einem Editor ihrer Wahl.

Am wichtigsten ist der Einsprungpunkt, der das Erzeugen einiger Membervariablen demonstriert. Unser Beispiel braucht ein Pipelineobjekt, für die Einzelelemente sind dann zwei Element-Handles erforderlich:

```
void app_main(void)
{
    audio_pipeline_handle_t pipeline;
    audio_element_handle_t i2s_stream_writer, mp3_decoder;
    ...
}
```

Die Inbetriebnahme der Hardware erfolgt im ADF unter Nutzung einer als Board bezeichneten Abstraktion. Ihre Auswahl beziehungsweise Parametrierung erfolgt im Rahmen von `Menuconfig`. Die im Code-Behind anzulegenden Befehle werten vor allem die Kompilationskonstanten aus:

```
audio_board_handle_t board_handle = audio_board_init();
audio_hal_ctrl_codec(board_handle->audio_hal,
    AUDIO_HAL_CODEC_MODE_BOTH, AUDIO_HAL_CTRL_START);
...
int player_volume;
audio_hal_get_volume(board_handle->audio_hal,
    &player_volume);
```

Interessant ist hier, dass der HAL auch Einstellungen zur Lautstärke-Regelung vorhält. Als Nächstes ist die Einrichtung des Haupt-Pipelineobjekts erforderlich:

```
audio_pipeline_cfg_t pipeline_cfg =
    DEFAULT_AUDIO_PIPELINE_CONFIG();
pipeline = audio_pipeline_init(&pipeline_cfg);
mem_assert(pipeline);
```

ESP-IDF-erfahrene Entwickler finden an dieser Stelle vor allem Bekanntes vor. Das Pipelineobjekt entsteht durch Übergeben eines Konfigurations-Structs, und kann danach von Verwender zu Verwender weitergereicht werden.

Darauf folgt die Erzeugung der eigentlichen Pipelineelemente. Zuerst an der Reihe ist der MP3-Decoder:

```
mp3_decoder_cfg_t mp3_cfg =
```

```
    DEFAULT_MP3_DECODER_CONFIG();
mp3_decoder = mp3_decoder_init(&mp3_cfg);
audio_element_set_read_cb(mp3_decoder, mp3_music_read_cb, NULL);
```

Hier stellt sich – insbesondere in Anbetracht der in Bild 3 gezeigten Pipeline – die Frage, wie die Datenbereitstellung erfolgt. Die Antwort darauf ist die Methode `audio_element_set_read_cb`, der eine nach folgendem Schema aufgebaute Funktion als Parameter übergeben wird:

```
int mp3_music_read_cb(audio_element_handle_t el,
    char *buf, int len,
    TickType_t wait_time, void *ctx)
{
    int read_size = file_marker.end
        - file_marker.start - file_marker.pos;
    if (read_size == 0) {
        return AEL_IO_DONE;
    } else if (len < read_size) {
        read_size = len;
    }
    memcpy(buf, file_marker.start +
        file_marker.pos, read_size);
    file_marker.pos += read_size;
    return read_size;
}
```

Über den Puffer liefert das Callback dabei die Informationen an, die für den Codec zu verarbeiten sind. Der I2S-Stream ist dann einfacher; er wird durch Übergeben von `AUDIO_STREAM_WRITER` als Ausgang parametrierung, und wirft die eingehenden Informationen in Richtung der I2S-Soundhardware:

```
i2s_stream_cfg_t i2s_cfg = I2S_STREAM_CFG_DEFAULT();
i2s_cfg.type = AUDIO_STREAM_WRITER;
i2s_stream_writer = i2s_stream_init(&i2s_cfg);
```

Das bloße Erzeugen von Pipeline-Elementen meldet diese nicht automatisch an: Dies ist ein Entgegenkommen von ESP-ADF an Entwickler, die mehrere Pipelines gleichzeitig nutzen wollen. Die eigentliche Einrichtung der Pipeline erfolgt stattdessen durch Anmelden der individuellen Teile, die je auch einen als ID dienenden String eingeschrieben bekommen:

```
audio_pipeline_register(pipeline, mp3_decoder, "mp3");
audio_pipeline_register(pipeline, i2s_stream_writer,
    "i2s");
const char *link_tag[2] = {"mp3", "i2s"};
audio_pipeline_link(pipeline, &link_tag[0], 2);
```

Im zweiten Akt folgt dann das Verlinken der Pipeline anhand eines Arrays, das die einzelnen String-IDs in korrekter Reihenfolge anliefert. Beim MP3-Player fehlt an dieser Stelle dann nur noch die Aktivierung der Pipeline:



### Listing 1. Event-Verarbeitung beim MP3-Player

```

while (1) {
    audio_event_iface_msg_t msg;
    esp_err_t ret = audio_event_iface_listen(evt, &msg, portMAX_DELAY);
    if (ret != ESP_OK) {
        continue;
    }
    if (msg.source_type == AUDIO_ELEMENT_TYPE_ELEMENT && msg.source == (void *) mp3_decoder
        && msg.cmd == AEL_MSG_CMD_REPORT_MUSIC_INFO) {
        audio_element_info_t music_info = ;
        audio_element_getinfo(mp3_decoder, &music_info);
        ESP_LOGI(TAG, "[ * ] Receive music info from mp3 decoder, sample_rates=%d, bits=%d, ch=%d",
            music_info.sample_rates, music_info.bits, music_info.channels);
        audio_element_setinfo(i2s_stream_writer, &music_info);
        i2s_stream_set_clk(i2s_stream_writer, music_info.sample_rates,
            music_info.bits, music_info.channels);
        continue;
    }
}
}

```

```

ESP_LOGI(TAG, "[ 5.1 ] Start audio_pipeline");
. . .
audio_pipeline_run(pipeline);

```

Im Fall des MP3-Players findet sich danach noch eine vergleichsweise umfangreiche Ereignisverarbeitung, die auf verschiedene Events reagiert (**Listing 1**). Die genaue Eventverarbeitung ist an dieser Stelle irrelevant – interessanter ist es, probeweise eine Kompilation anzustoßen:

```

(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/examples/get-started/play_mp3_control$ make build

```

Hierzu ein Hinweis: Der Autor verfasst diesen Artikel – im Interesse der Separation zu Kundendaten – auf seinem Reiselaptop. Die dortige Ubuntu-Version bringt CMake in Version 3.4.3 mit, weshalb die Ausführung von *idf.py* nicht möglich ist. Im praktischen Einsatz ist die Nutzung von *idf.py* der Verwendung von *make* natürlich immer vorzuziehen und in der aktuellen Version IDF-Version 5.x sogar verpflichtend.

Lohn der Mühen ist bei einem jungfräulichen Projektskelett das Aufscheinen des *Menuconfig*-Bildschirms, der gegenüber einem normalen IDF-Projekt einige Zusatzoptionen anbietet. Am Wichtigsten ist die Option (Top)→Audio→HAL→Audio board, wo Sie wie in **Bild 5** gezeigt die passende Board-Konfiguration auswählen dürfen.

Nach dem Speichern beginnt der von gewöhnlichen IDF-Projekten bekannte Buildprozess, auch die Auslieferung erfolgt – ESP32 ist nun mal ESP32 – wie gewohnt. Wichtig ist dabei vor allem, dass die Platine zwei Micro-USB-Ports aufweist: Power kümmert sich ausschließlich um die Energieversorgung, während UART nur für die Datenkommunikation zuständig ist. Der Autor löst dieses Problem gern dadurch, dass er einfach zwei Kabel zwischen Workstation und Board schaltet.

Wichtig ist, dass der LyraT – anders als andere Entwicklerboards –

nicht automatisch neu programmiert werden kann. Schon vor dem Aufscheinen des „Connecting“-Abschnitts sollten Sie das Board in den Entwicklermodus versetzen. Hierzu halten Sie Boot gedrückt, drücken dann kurz RST und lassen den Bootknopf erst danach los:

```

Serial port /dev/ttyUSB0
Connecting.....

```

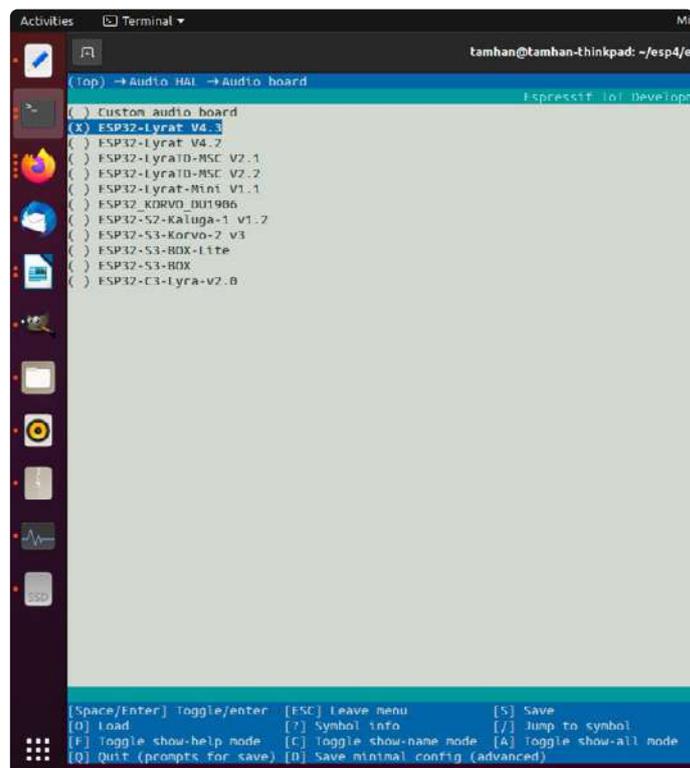


Bild 5: ADF erweitert Menuconfig um einige Einstellungen.

Wer diesen Affengriff nicht korrekt hinbekommt, wird mit der folgenden – verwirrenden – Fehlermeldung abgespeist:

```
A fatal error occurred: Failed to connect to ESP32:
Invalid head of packet (0x1B): Possible serial noise or
corruption.
```

Nach dem erfolgreichen Durchlaufen des Flashprozesses stecken Sie Kopfhörer in die Phone-Buchse und drücken abermals auf *Reset*. Lohn der Mühen ist das Abspielen eines kleinen Lieds.

Zu guter Letzt sei noch darauf hingewiesen, dass die Ausführung des Konfigurationsskripts *export.sh* nicht ausreicht, um die Umgebung komplett zu parametrieren. Vor seiner Aktivierung ist es immer erforderlich, die Umgebungsvariable *ADF\_PATH* korrekt einzurichten:

```
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf$ export
ADF_PATH=$PWD
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/esp-idf$
./export.sh
Setting IDF_PATH to '/home/tamhan/esp4/esp-adf/esp-idf'
...
```

## Verarbeitung von Eingangsdaten

Die Nutzung der ESP-ADF-Pipeline hat nicht nur im Rahmen der Medienwiedergabe einen Sinn: Es ist genauso legitim, Berechnungen in der Pipeline unterzubringen. Ein Kundenprojekt des Autors fütterte beispielsweise einen Algorithmus mit Mikrofondaten. Die eigentliche Berechnung kam dabei in einem Pipelineelement unter, dessen generelle Struktur wir hier kurz vorstellen wollen. Beginnen wir mit der Initialisierung der Pipeline: Da die Audioinformationen über den I2S-Bus eintreffen, ist im ersten Schritt abermals ein I2S-Stream-Element erforderlich. Seine Konfiguration nimmt nun aber den Flag *AUDIO\_STREAM\_READER* auf, der es als Eingang beziehungsweise Datenquelle markiert:

```
i2s_stream_cfg_t i2s_cfg = I2S_STREAM_CFG_DEFAULT();
i2s_cfg.type = AUDIO_STREAM_READER;
#if defined CONFIG_ESP_LYRAT_MINI_V1_1_BOARD
    i2s_cfg.i2s_port = 1;
```

```
#endif
i2s_stream_reader = i2s_stream_init(&i2s_cfg);
```

Neben dem I2S-Stream benötigen wir auch seinen Kollegen, der den eigentlichen proprietären Algorithmus enkapsulieren wird. Hier seine Initialisierung:

```
tams_stream_cfg_t fatfs_cfg = TAMS_STREAM_CFG_DEFAULT();
fatfs_cfg.type = AUDIO_STREAM_WRITER;
tams_stream = tams_stream_init(&fatfs_cfg);
```

Die eigentliche Errichtung der Pipeline erfolgt dann durch Zuweisung von Strings und deren Weiterreichung an *audio\_pipeline\_link*:

```
audio_pipeline_register(pipeline, i2s_stream_reader,
    "i2s");
audio_pipeline_register(pipeline, tams_stream, "tam");
audio_pipeline_link(pipeline, (const char *[]) {"i2s",
    "tam"}, 2);
```

Damit können wir in die Headerdatei wechseln, die die für die Inbetriebnahme des „Tam“-Tasks notwendigen Elemente bereitstellt. Am wichtigsten ist der Konfigurations-Struct, der unter anderem die für FreeRTOS notwendigen Metainformationen enthält (**Listing 2**). *TAMS\_STREAM\_CFG\_DEFAULT* ist dann ein Convenience-Makro, das die Erzeugung der Structs mit Standardparametern ermöglicht. Aus Platzgründen können wir die Konstanten hier nicht abdrucken:

```
#define TAMS_STREAM_CFG_DEFAULT() {\
    .task_prio = TAMS_STREAM_TASK_PRIO, \
    . . .
}
```

Die einzige Ausnahme machen wir für *BUF\_SIZE*. Die angelieferten Daten sind 16 Bit breit, weshalb wir zum Bearbeiten von 1024 Slots langen Blöcken folgenden Konstantenwert benötigen:

```
#define TAMS_STREAM_BUF_SIZE (1024*2)
```



### Listing 2. Konfiguration des Pipeline-Elements

```
typedef struct {
    audio_stream_type_t    type;          /*!< Stream type */
    int                    buf_sz;        /*!< Audio Element Buffer size */
    int                    out_rb_size;   /*!< Size of output ringbuffer */
    int                    task_stack;    /*!< Task stack size */
    int                    task_core;     /*!< Task running in core (0 or 1) */
    int                    task_prio;     /*!< Task priority (based on freeRTOS priority) */
} tams_stream_cfg_t;
```



### Listing 3. Initialisierung

```
audio_element_handle_t tams_stream_init(tams_stream_cfg_t *config)
{
    audio_element_handle_t el;
    tams_stream_t *fatfs = audio_malloc(1, sizeof(tams_stream_t));

    AUDIO_MEM_CHECK(TAG, fatfs, return NULL);

    audio_element_cfg_t cfg = DEFAULT_AUDIO_ELEMENT_CONFIG();
    cfg.open = _tams_open;
    cfg.close = _tams_close;
    cfg.process = _tams_process;
    cfg.destroy = _tams_destroy;
    cfg.task_stack = config->task_stack;
    cfg.task_prio = config->task_prio;
    cfg.task_core = config->task_core;
    cfg.out_rb_size = config->out_rb_size;
    cfg.buffer_len = config->buf_sz;
    if (cfg.buffer_len == 0) {
        cfg.buffer_len = TAMS_STREAM_BUF_SIZE;
    }
    cfg.tag = "file";
    ...
}
```

Damit können wir uns der Initialisierung zuwenden, siehe **Listing 3**.

Der Wert `cfg.buffer_len` ist dabei wichtig, weil er die Menge der pro Durchlauf anzuliefernden Datenworte festlegt. Ansonsten findet sich hier im Wesentlichen nur Boilerplatecode, der Housekeeping erledigt:

```
if (config->type == AUDIO_STREAM_WRITER) {
    cfg.write = _tams_write;
} else {
    cfg.read = _tams_read;
}
el = audio_element_init(&cfg);

AUDIO_MEM_CHECK(TAG, el, goto _tams_init_exit);
audio_element_setdata(el, fatfs);
return el;

_tams_init_exit:
audio_free(fatfs);
return NULL;
```

Vom Entwickler erzeugte Audioelemente müssen sich in eine wie in Bild 3 gezeigte Pipeline integrieren. `audio_element_init` benötigt dazu einige Informationen – neben den threadbezogenen Parametern richtet die Routine auch eine Gruppe von Callbacks ein, über die das Audioframework das Element über auftretende Ereignisse informieren kann.

Als Beispiel wollen wir uns die Methode `tams_open` ansehen, die sich nach folgendem Schema um die Initialisierung des Elements kümmert:

```
static esp_err_t _tams_open(audio_element_
handle_t self) {
    audio_element_info_t info;
    audio_element_getinfo(self, &info);

    initTamsWorkerAlgo();

    return audio_element_setinfo(self,
        &info);
}
```

Da der vorliegende Algorithmus die in `audio_element_info_t` befindlichen Informationen nicht benötigt, beschränkt sich die Implementierung auf das Weiterreichen der Parameterinformationen an die Pipeline.

Für den Datenaustausch sind die Ereignisse `read` und `write` verantwortlich. Da unser Stream ausschließlich als „Schreibstelle“ dient und nicht auslesbar ist, beschränkt er sich bei einem Ausleserversuch auf das Ausgeben einer Fehlermeldung in das Systemlog:

```
static int _tams_read(audio_element_
handle_t self,
char *buffer, int len,
TickType_t ticks_to_wait, void *context)
{
    ESP_LOGE(TAG, "CENSORED");
    return 0;
}
```

Schreiboperationen sind ebenfalls als „Passthrough“ implementiert, führen aber keine Operationen aus:

```
static int _tams_write(audio_element_handle_t self,
char *buffer, int len,
TickType_t ticks_to_wait, void *context)
{
    return len;
}
```

Im nächsten Schritt ist ein nach folgendem Schema aufgebauter Eingangspuffer erforderlich:

```
int16_t DspBuf[4096];
```

`_tams_process` ist dann für die eigentliche Datenverarbeitung verantwortlich. Am wichtigsten ist hier der Aufruf von `audio_element_input`, der sich um das Umkopieren der Informationen in Richtung des Arbeitspuffers kümmert:

```
static int _tams_process(audio_element_handle_t self,
char *in_buffer, int in_len) {
    audio_element_input(self, (char *)DspBuf, in_len);
}
```

```

tamhan@tamhan-thinkpad:
├── document
├── main
├── Makefile
├── README_CN.md
├── README.md
├── pipeline_passthru
│   ├── CMakeLists.txt
│   ├── main
│   ├── Makefile
│   ├── README_CN.md
│   └── README.md
├── pipeline_resample
│   ├── CMakeLists.txt
│   ├── main
│   ├── Makefile
│   ├── README_CN.md
│   └── README.md
├── pipeline_sonic
│   ├── CMakeLists.txt
│   ├── main
│   ├── Makefile
│   ├── README_CN.md
│   └── README.md
├── pipeline_spliffs_anr_resample
│   ├── CMakeLists.txt
│   ├── main
│   ├── Makefile
│   ├── partitions_spliffs_example.csv
│   ├── README_CN.md
│   ├── README.md
│   └── sdkconfig.defaults
└── 15 directories, 32 files
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/examples/audio_processing$ tree -
├── pipeline_aic
├── pipeline_audio_forge
├── pipeline_equalizer
├── pipeline_passthru
├── pipeline_resample
├── pipeline_sonic
└── pipeline_spliffs_anr_resample

```

Bild 6: Im Pipeline-Zoo, da wimmelt es!

Als Beispiel für die Datenverarbeitung führt der Autor an dieser Stelle eine Normalisierung des Wertebereichs vor:

```

for (int i=0 ; i< in_len ; i++) {
    y_cf[i] = ((float)DspBuf[i]) / (float)32768;;
}

```

Zu guter Letzt ist noch Housekeeping erforderlich:

```

int r_size = audio_element_input(self, in_buffer,
    in_len);
int w_size = 0;
if (r_size > 0) {
    w_size = audio_element_output(self, in_buffer,
        r_size);
} else {
    w_size = r_size;
}
return w_size;
}

```

Für das Abtragen des Streams sind vor allem Operationen erforderlich, die den „geblockten“ Speicher freigeben:

```

static esp_err_t _tams_close(audio_element_handle_t self)
{
    return ESP_OK;
}
static esp_err_t _tams_destroy(audio_element_handle_t
self) {
    tams_stream_t *fatfs =
    (tams_stream_t *)audio_element_getdata(self);
}

```

```

audio_free(fatfs);
return ESP_OK;
}

```

Der Rest des Arbeitscodes in der Lösung des Autors hat dann nur die Aufgabe, nach folgendem Schema ein Anhalten des Kerns zu bewirken:

```

audio_pipeline_run(pipeline);
printf("halting.\n");
for(;;);

```

Beachten Sie, dass die in der Pipeline periodisch aufgerufene Arbeiterfunktion im Prinzip tun und lassen kann, was immer sie will – im Projekt des Autors wird als Reaktion auf eingehende Sounddaten je nach Konfiguration entweder mit I<sup>2</sup>C- oder mit SPI-Gegenstellen kommuniziert.

### Mehr Pipelineelemente

Unter `~/esp4/esp-adf/examples/audio_processing` stellt ESP-ADF die in **Bild 6** gezeigte Gruppe zusätzlicher Beispiele zur Verfügung – die in ihnen implementierten Algorithmen realisieren durch die Bank zusätzliche Pipelineoperationen.

Interessant empfand der Autor das Beispiel `examples/audio_processing/pipeline_equalizer`, da es einen vollständigen grafischen Equalizer realisiert. Er benutzt eine im ESP-ADF-Framework enthaltene Klasse und kommt deshalb ohne fortgeschrittene Mathematik aus. Stattdessen erfolgt die Inbetriebnahme nach folgendem bekannten Schema:

```

equalizer_cfg_t eq_cfg = DEFAULT_EQUALIZER_CONFIG();
int set_gain[] = { -13, -13, -13, -13, -13, -13, -13,
-13, -13, -13, -13, -13, -13, -13, -13, -13, -13,
-13, -13};
eq_cfg.set_gain = set_gain;
equalizer = equalizer_init(&eq_cfg);

```

Mehr über Datenfelder erfährt man unter [4].

Die eigentliche Einbindung in den Wiedergabestrom erfolgt dann wie gewohnt mittels `audio_pipeline_register`:

```

audio_pipeline_register(pipeline, fatfs_stream_reader,
    "file_read");
audio_pipeline_register(pipeline, wav_decoder, "wavdec");
audio_pipeline_register(pipeline, equalizer,
    "equalizer");
audio_pipeline_register(pipeline, i2s_stream_writer,
    "i2s");

```

Ab einem gewissen Komplexitätsgrad des Systems führt kein wirklich gangbarer Weg an hauseigenen Signalverarbeitungselementen vorbei. Espressif hilft Entwicklern an dieser Stelle mit der unter [5] bereitstehenden Bibliothek ESP-DSP aus: eine Art Framework, das verschiedene DSP-Algorithmen in hoher algorithmischer Stabilität und mit diversen Optimierungen für die verschiedenen bei Espressif geläufigen Chips mitbringt.



Da jedes ESP32-Projekt vom Aufbau der Ordnerstruktur her direkt für die Einbindung zusätzlicher ESP-IDF-Komponenten ausgerüstet ist, ist die Installation von DSP einfach. Erzeugen Sie im ersten Schritt eine Kopie eines Projekts:

```
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/examples$  
cp -r audio_processing/pipeline_equalizer/ tamsdspstest1  
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/examples$  
cd tamsdspstest1/
```

Legen sie dort einen Ordner mit dem magischen Namen *components* an. Dieser nimmt danach die aus GitHub herunterladbare Vollversion der Bibliothek entgegen:

```
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/examples/  
tamsdspstest1$ mkdir components  
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/examples/  
tamsdspstest1$ cd components/  
(base) tamhan@tamhan-thinkpad:~/esp4/esp-adf/examples/  
tamsdspstest1/components$ git clone https://github.com/  
espressif/esp-dsp.git
```

Ab dem nächsten Durchlauf der Kompilationsstoolchain sehen Sie – wie in **Bild 7** gezeigt – das Aufscheinen einer neuen Option zur Konfiguration des Verhaltens der DSP-Bibliothek.

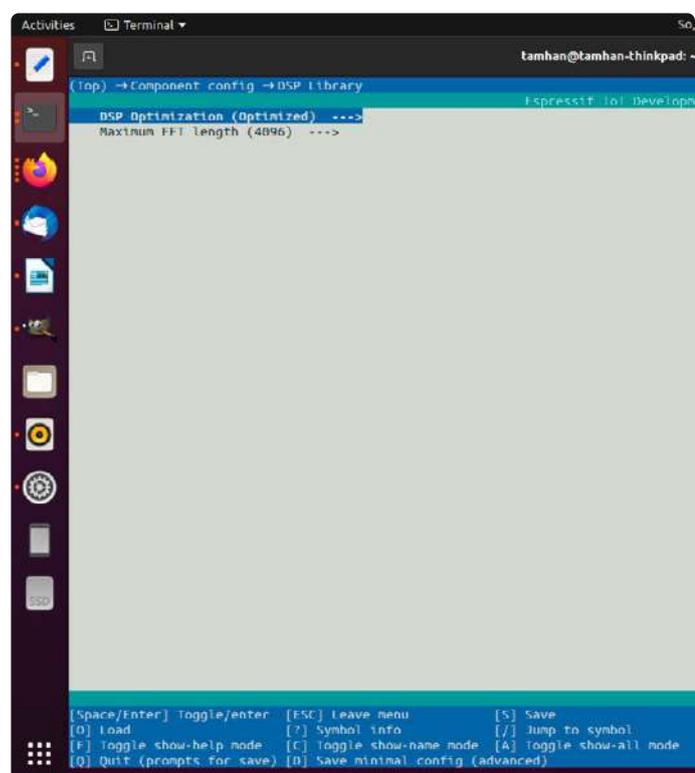


Bild 7: ESP-DSP nistet sich unter (Top)→Component config→DSP Library in den Kompilationsprozess ein.

## Fazit

Mit ESP-ADF stellt Espressif Entwicklern ein leistungsfähiges und flexibles Framework zur Realisierung von Audioanwendungen zur Verfügung. Im Unternehmen des Autors sparte seine Verwendung Dutzende, wenn nicht sogar hunderte Mannstunden ein – auf jeden Fall empfehlenswert! ◀

220600-02

## Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Dann kontaktieren Sie bitte den Autor unter [tamhan@tamoggemon.com](mailto:tamhan@tamoggemon.com) oder die Elektor-Redaktion unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

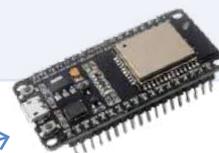
## Über den Autor

Ingenieur Tam Hanna befasst sich seit mehr als 20 Jahren mit Elektronik, Computern und Software; er ist freiberuflicher Entwickler, Buchautor und Journalist ([www.instagram.com/tam.hanna](https://www.instagram.com/tam.hanna)). In seiner Freizeit beschäftigt sich Tam unter anderem mit 3D-Druck und dem Vertrieb von Zigarren.



## Passendes Produkt

➤ **ESP32-DevKitC-32D**  
[www.elektor.de/18701](http://www.elektor.de/18701)

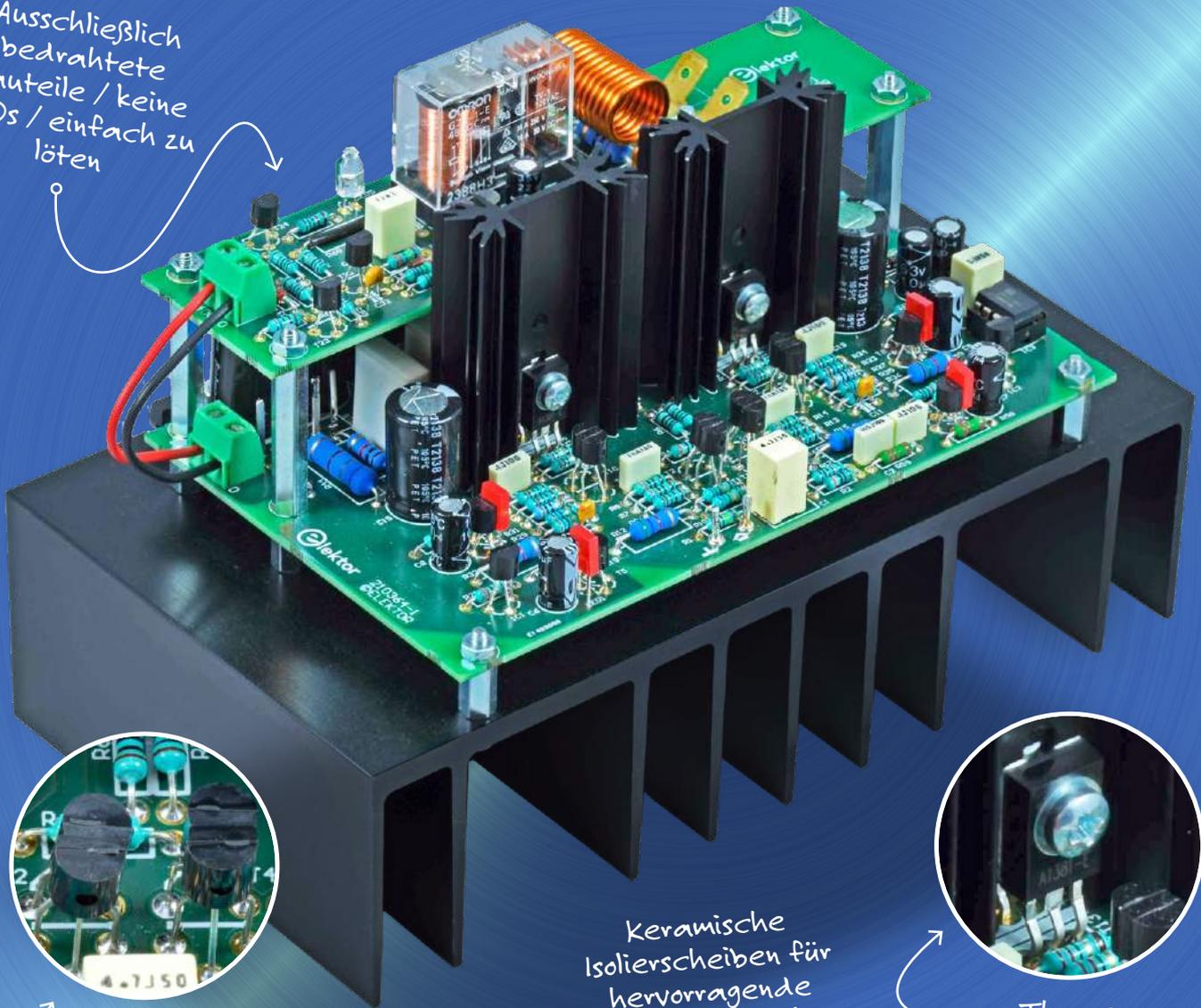


## WEBLINKS

- [1] ESP-ADF Peripherals : [https://espressif-docs.readthedocs-hosted.com/projects/esp-ADF/en/latest/api-reference/peripherals/esp\\_peripherals.html](https://espressif-docs.readthedocs-hosted.com/projects/esp-ADF/en/latest/api-reference/peripherals/esp_peripherals.html)
- [2] ESP-ADF Button Peripheral: [https://espressif-docs.readthedocs-hosted.com/projects/esp-ADF/en/latest/api-reference/peripherals/periph\\_button.html](https://espressif-docs.readthedocs-hosted.com/projects/esp-ADF/en/latest/api-reference/peripherals/periph_button.html)
- [3] ESP-ADF API Reference: <https://espressif-docs.readthedocs-hosted.com/projects/esp-ADF/en/latest/api-reference/index.html>
- [4] ESP-ADF Equalizer: <https://espressif-docs.readthedocs-hosted.com/projects/esp-ADF/en/latest/api-reference/audio-processing/equalizer.html>
- [5] ESP-DSP: <https://github.com/espressif/esp-dsp>
- [6] ESP32-LyraT Getting Started Guide: <https://espressif-docs.readthedocs-hosted.com/projects/esp-ADF/en/latest/design-guide/dev-boards/get-started-esp32-lyrat.html>

# Elektor-Leistungsverstärker- Bausatz Fortissimo-100

Ausschließlich  
bedrahtete  
Bauteile / keine  
SMDs / einfach zu  
löten



Kleinsignal-  
Transistoren BC546B  
und BC556B

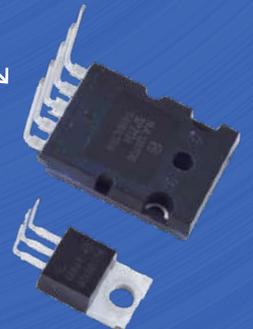


Thermische  
Entlastung der  
Transistoranschlüsse in  
der Gegentaktstufe



TO-264-5-  
Leistungstransistoren und  
TO-220-Treiber unterhalb  
der Platine auf Kühlkörper  
angebracht

Ausgangstransistoren  
mit gewinkelten  
Anschlüssen





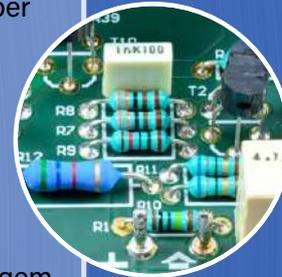
## Der Entwurf

- › Voll analoger Klasse-AB-Verstärker
- › 100 % symmetrisch
- › Extrem niedrige Verzerrungen und vernachlässigbares Rauschen
- › ThermalTrak-Leistungstransistoren machen die Ruhestromeinstellung der Endtransistoren überflüssig
- › Symmetrisches Bootstrapping für maximalen Ausgangshub
- › Geregelte  $\pm 40\text{-V}$ -Stromversorgung
- › Minimale Lastimpedanz  $3\ \Omega$
- › Solide, kompakte Platine auf Kühlkörper montiert
- › Mechanische Arbeiten beschränken sich auf zwölf M3-Bohrungen beim Kühlkörper
- › Erkennung der Versorgungsspannung
- › Schutz gegen Gleichspannung am Ausgang



## Die Einzelteile

- › Elektor-Bausatz: Platinen + alle Bauteile (bedrahtet) + Kühlkörper
- › ThermalTrak-Transistoren mit integrierten Vorspannungsdioden
- › Treiber MJE15023 und MJE15033
- › Gegentakt-Spannungsverstärkung mit KSC3503 und KSA1381 für höchste Linearität
- › DC-Regelkreis mit OPA177 (DIP-8)
- › 1%-Metallfilmwiderstände (0,6 W) in allen Schaltungsteilen niedriger Leistung
- › Langlebige Hochtemperatur-Elektrolytkondensatoren mit niedrigem ESR-Wert für Bootstrap und Versorgungsspannungsentkopplung



## Eigenschaften

Eingangsempfindlichkeit	1,076 V (94 W / $8\ \Omega$ , THD = 0,1 %, B = 22 kHz)
Eingangsimpedanz	10 k $\Omega$
Dauerleistung	94 W ( $8\ \Omega$ , THD = 0,1 %) 181 W ( $4\ \Omega$ , THD = 0,1 %)
Bandbreite	3,3 Hz ... 237 kHz (-3 dB, 1 W/ $8\ \Omega$ )
Leerlaufbandbreite	$\approx 20$ kHz
Open-Loop-Verstärkung	$\approx 140.000$ ( $8\ \Omega$ load)
Anstiegsgeschwindigkeit	45 V/ $\mu\text{s}$
Signal-Rausch-Verhältnis	103 dB (B = 22 Hz - 22 kHz linear)
THD plus Rauschen	0,0008 % (1 kHz, 50 W, $8\ \Omega$ , B = 80 kHz) 0,002 % (20 kHz, 50 W, $8\ \Omega$ , B = 80 kHz) 0,0042 % (20 kHz, 100 W, $4\ \Omega$ , B = 80 kHz)
Intermodulationsverzerrung (50 Hz : 7 kHz = 4 : 1)	0,0015 % (50 W, $8\ \Omega$ ) 0,0041 % (100W, $4\ \Omega$ )

**Mehr Infos**

[www.elektor.de/20273](http://www.elektor.de/20273)



# Licht für Klangeffekte nutzen

## LDR-basiertes spannungsgesteuertes 24 dB/Okt.-Synthesizer-Filter

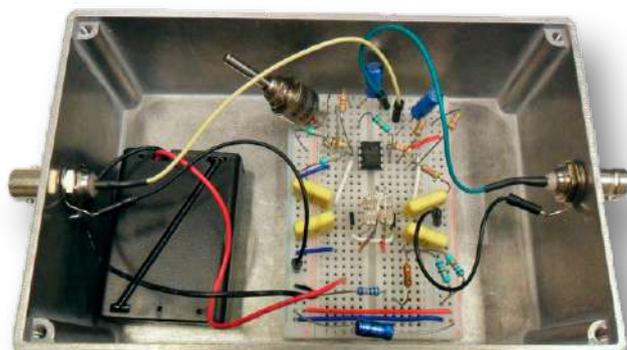
Von Raymond Schouten (Niederlande)

Dieser Artikel beschreibt eine einfache abstimmbare Filterschaltung, die keine speziellen Chips benötigt, sondern nur zwei Operationsverstärker, die mit einfachen +5 V betrieben werden können. Trotz seiner Einfachheit glänzt das Filter durch sehr geringe Verzerrungen (THD < 0,01%), gute Großsignalverarbeitung und einen weiten Dynamikbereich (> 90 dB).

Das hier vorgestellte Audio-Tiefpassfilter mit einer Steilheit von 24 dB/Oktave basiert auf vier lichtabhängigen Widerständen, besser bekannt als LDRs, die von einer einzigen LED gesteuert werden. Die Helligkeit der LED steuert die Grenzfrequenz des Filters von etwa 20 Hz bis zu 20 kHz. Die Dämpfung (oder Q-Faktor oder Güte(faktor) oder Resonanzschärfe, es gibt viele Bezeichnungen für ein und dieselbe Sache) kann bis zur vollen Schwingung eingestellt werden, was für Musiksynthesizer-Anwendungen interessant ist. Die Schaltung ist

einfach zu bauen, und die Bauteilkosten liegen bei nur etwa 4 €.

Dieses Filter wurde mit Blick auf Musiksynthesizer entwickelt, aber seine hohe Signalqualität könnte auch für andere Audioanwendungen attraktiv sein. Die Schaltung wurde zunächst simuliert, bevor ein Prototyp auf einem Breadboard gebaut wurde, um die Leistung des Filters zu abzuschätzen. Klangbeispiele und ein Demo-Video sind auf der Elektor-Labs-Projektseite [1] verfügbar.



*Bild 1. Die Single-Supply-Version des Filters wurde auf einer Lochrasterplatine aufgebaut und zur Prüfung seiner Eigenschaften in eine Metallbox eingebaut, während sie von drei AAA-Batterien versorgt wurde.*

### Eine Schaltung - zwei Versionen

Es gibt zwei Versionen des Filters, von denen die einfachere mit einer symmetrischen (dualen) Versorgung arbeitet. Durch Hinzufügen einiger Widerstände und Kondensatoren kann es in die Version umgewandelt werden, die mit einer einfachen Versorgungsspannung funktioniert. Diese zweite Version wurde gebaut und gemessen (**Bild 1**).

Wir wenden uns allerdings zunächst der Schaltung des Filters mit symmetrischer Versorgung in **Bild 2** zu, da weniger Bauteile einfacher zu erläutern sind. Die Version mit einfacher Versorgung in **Bild 3** funktioniert aber genau gleich.

Das Herzstück der Schaltung ist ein RC-Tiefpassfilter mit einer Flankensteilheit von 6 dB/Oktave, bei dem der Festwiderstand R durch einen LDR ersetzt ist. Vier dieser Filter werden in Reihe geschaltet, um eine Flankensteilheit von 24 dB/Oktave zu erreichen. Die gleichzeitige Steuerung des Widerstands der vier LDRs durch eine einzige LED führt zu einem Filter mit einer Grenzfrequenz, die in einem weiten Bereich (> 1:1000) eingestellt werden kann.

Die Kaskadierung von vier identischen RC-Schaltkreisen ohne Puffer mag wie ein Rezept für ein Desaster aussehen, das zu einem sehr schlechten Filter mit schlampigem Übergang bei der Grenzfrequenz führt, aber da das berühmte Moog-Leiterfilter auf genau diesem Prinzip basiert, kann es ganz so schlecht nicht sein. Robert Moog hat das Problem auf clevere Weise gelöst, eine Lösung, die auch in unserer Schaltung verwendet wird. Sie basiert auf einer

negativen Über-alles-Rückkopplung, die erst bei der Grenzfrequenz in eine positive Rückkopplung übergeht. Im Kasten *Steuerung der Filtergüte* wird dies näher erläutert. Der Opamp U1a puffert den Eingang und fügt das Rückkopplungssignal über R9 und P1 hinzu. Der Operationsverstärker U1b ist ein Puffer mit hoher Impedanz für die LDR/C-Kaskade und fügt auch einen Teil der Verstärkung hinzu, um den Verlust in der Nähe der Grenzfrequenz zu kompensieren. P1 steuert die Stärke der negativen Über-alles-Rückkopplung. Durch Drehen des Potis von 100 kΩ auf 0 Ω wird das Filter von gedämpfter Reaktion auf volle Oszillation gestellt. Ersetzt man P1 durch eine weitere LED-LDR-Kombination, kann der Q-Faktor auch elektronisch gesteuert werden. Die Frequenzabstimmung und die Kurven der Gütesteuerung sind im Kasten mit dem Titel *Simulation der Filterantwort* zu finden. Eine ausführliche Erläuterung, wie man LED D1 auswählt, und des Konzepts der zugehörigen Widerstandsschaltung finden Sie im Kasten *Die LED-LDR-Kombination*.

### Vorteile des Konzepts

Die meisten analogen abstimmbaren Filter verwenden komplexere Schaltungen, die nur bei kleinen Signalpegeln ( $< 100 \text{ mV}_{\text{SS}}$ ) funktionieren, um die Verzerrung auf einem akzeptablen Niveau zu

halten. Beispiele hierfür sind Filter, die auf Transistoren oder Dioden mit abgestimmter Vorspannung beruhen, die diskret oder in speziellen ICs (OTAs wie dem berühmten LM13700) integriert sind. Dieses LDR-basierte Design hingegen unterstützt Signale mit einer Amplitude von mehreren Volt bei sehr niedrigen Verzerrungswerten. Der Spannungshub wird hauptsächlich durch die Operationsverstärker und ihre Versorgungsspannung begrenzt, da die meisten LDRs bis zu 100 V oder mehr klaglos verkraften. Ja, man könnte dieses Filter sogar mit Röhren bauen! Da der Rausch-

pegel der Schaltung relativ niedrig liegt - er wird hauptsächlich durch die Operationsverstärker und das Äquivalent-Widerstandsrauschen der LDRs bestimmt - kann ein großer Dynamikbereich erzielt werden. Ein weiterer Vorteil der Schaltung ist die extrem geringe (oder nicht vorhandene?) Steuerspannungsleckage in den Audiopfad, da dieser nur optisch gekoppelt ist. Mit geeigneten Operationsverstärkern kann die Schaltung mit einer einzigen niedrigen Versorgungsspannung (3 V bis 5 V) betrieben werden, was sie mit Mikrocontrolleranwendungen wie Arduino kompatibel macht.

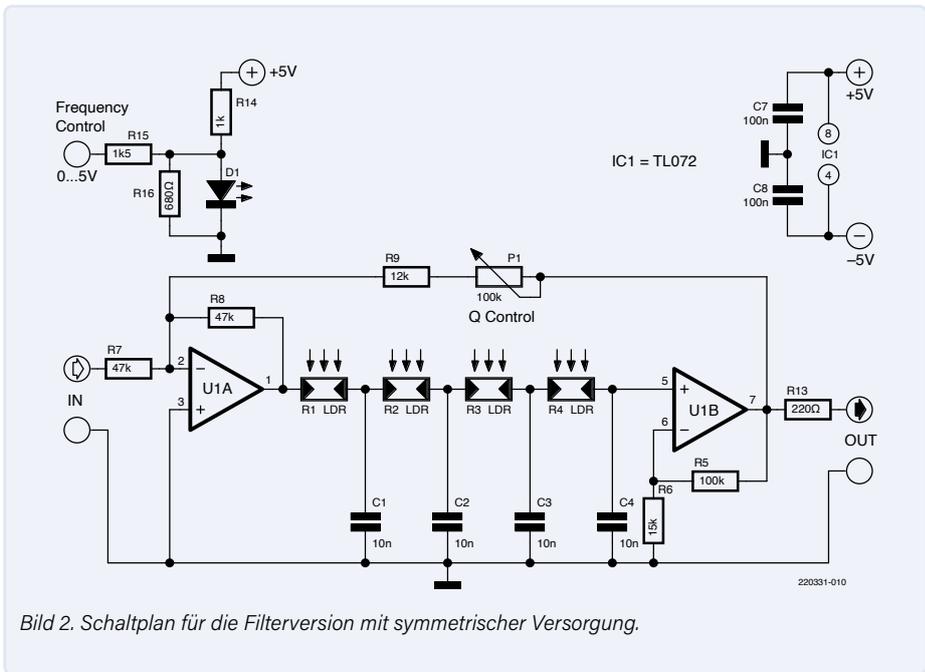


Bild 2. Schaltplan für die Filterversion mit symmetrischer Versorgung.

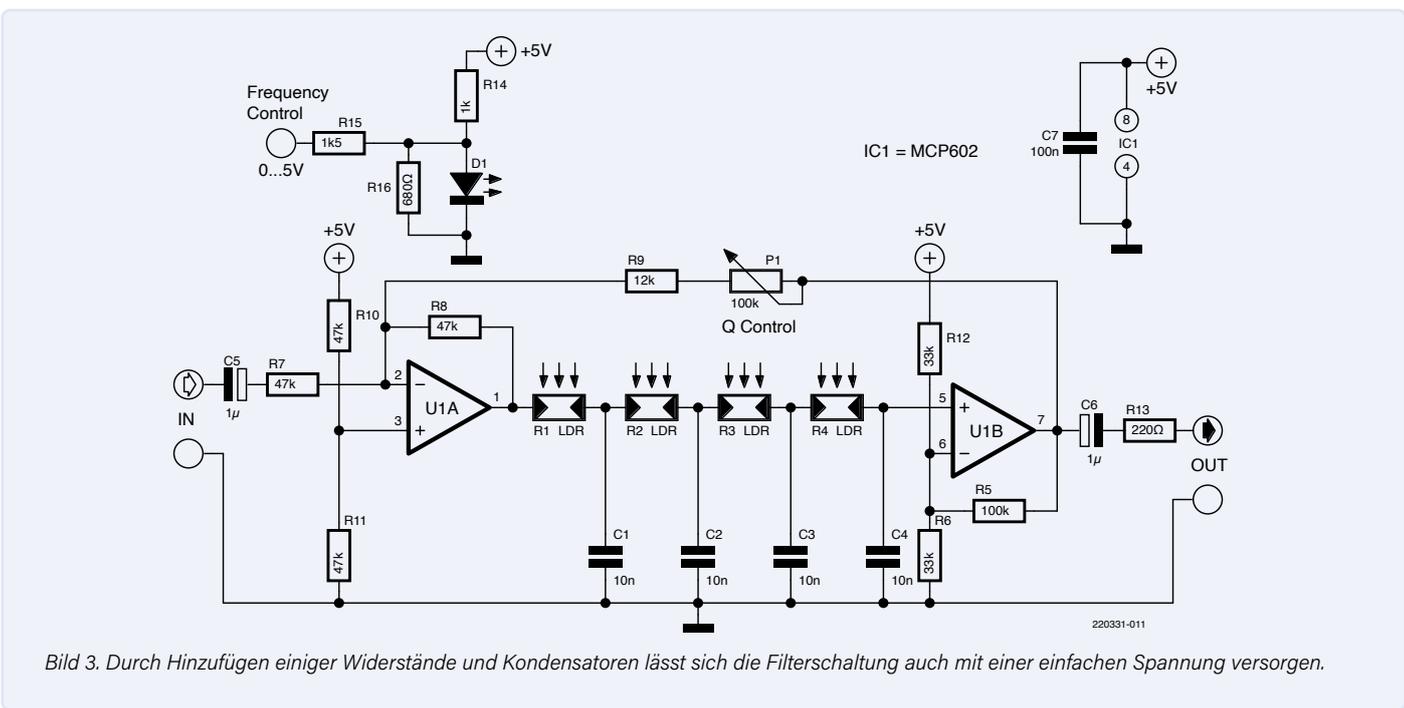


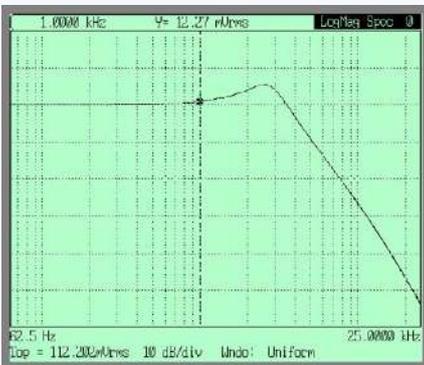
Bild 3. Durch Hinzufügen einiger Widerstände und Kondensatoren lässt sich die Filterschaltung auch mit einer einfachen Spannung versorgen.



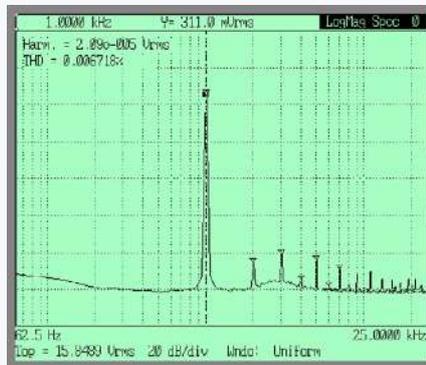
## Verzerrungsmessungen

Für diese Messungen wurde ein Spektralanalysator SR770 FFT der Marke Stanford Research mit eingebauter Signalquelle verwendet. Um den Einfluss elektrischer und optischer Störungen „von außen“ zu verhindern, wurde die Schaltung in einem geschlossenen Metallgehäuse untergebracht, das auch die Batterieversorgung enthält (siehe Bild 1). Die Schaltung wurde mit einer einfachen 4,5-V-Versorgung (drei AAA-Zellen) betrieben.

Zunächst wurde mit dem Analysator die Filterkennlinie ermittelt. Dazu erzeugte er eine 1-kHz-Sinuswelle und zeichnete das Spektrum des Filterausgangs bis 25 kHz auf. Daraus wurde die gesamte harmonische Verzerrung (THD) für die ersten sechs Harmonischen berechnet. Das Filter wurde dazu auf eine Grenzfrequenz von 4 kHz eingestellt. Der Grund für diese Wahl ist, dass es bei der Grenzfrequenz eine positive Rückkopplung gibt, die die Verzerrung erhöht. Dies ist also ein Worst-Case-Szenario für die Verzerrungen der (normalerweise dominanten) zweiten und dritten Harmonischen. Man könnte zwar dagegen argumentieren, dass ein Teil der höheren Oberwellen unterdrückt wird, aber eine Überprüfung bei  $f_c = 13$  kHz ergab fast das gleiche Oberwellenmuster.



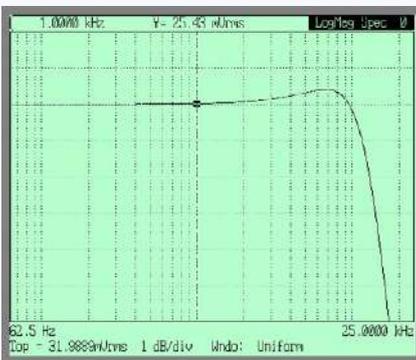
Gemessene Filterantwort.



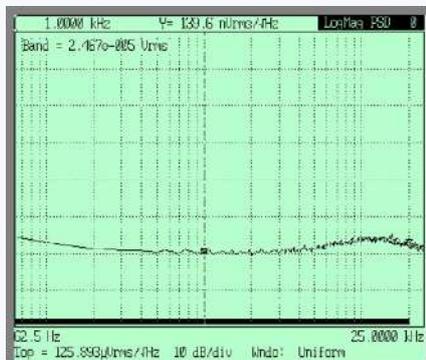
Gemessenes Verzerrungsspektrum des Filters.

## Rauschmessung und Dynamikbereich

Das Ausgangsrauschspektrum wird bei ausgeschalteter Quelle gemessen und zeigt insgesamt  $25 \mu V_{\text{eff}}$  im Audioband bis 20 kHz. In Kombination mit der maximalen Signalhöhe von  $4 V_{\text{ss}}$  ( $1,4 V_{\text{eff}}$ ) führt dies zu einem Dynamikbereich von 95 dB. Dieser Wert lässt sich durch eine höhere Versorgungsspannung weiter verbessern.



Gemessene Filterkennlinie (1 dB/Div). Das Filter wurde bei 13 kHz auf -3 dB eingestellt.



Gemessenes Rauschspektrum am Filterausgang.

Die Spannungssteuerung der LED (siehe Kasten *Die LED-LDR-Kombination*) ermöglicht eine etwas plumpe, aber irgendwie exponentielle Steuerkurve, wie sie für Synthesizer-Anwendungen benötigt wird. Für eine präzisere (aber lineare) Steuerung könnte eine Stromquelle oder ein PWM-Signal die Helligkeit der LED steuern.

## Beschränkungen des Konzepts

Fairerweise muss gesagt werden, dass das hier vorgestellte Filterkonzept im Vergleich zu den anderen Konzepten auch einige Nachteile aufweist. Die Einstellung der Grenzfrequenz ist nicht sehr präzise. Für einfache Synthesizeranwendungen ist sie zwar ausreichend, aber für genauere Anwendungen könnte eine Kalibriertabelle oder eine zusätzliche Rückkopplungsschaltung erforderlich sein.

Außerdem ist die Regelgeschwindigkeit der Grenzfrequenz durch die relativ langsamen LDRs begrenzt. Wenn die Filterfrequenz erhöht wird, bleibt die gemessene Reaktion in der Größenordnung von einer Millisekunde, aber bei Verringerung kann die Reaktion bis zu 50 ms und sogar bis zu 500 ms bei sehr niedrigen Grenzfrequenzen auf sich warten lassen. Kurze „Attack“-Zeiten sind daher möglich, um das Filter zu Beginn einer Note zu öffnen. In der Musik ist die Einschwingzeit eines Tons aber in der Regel kürzer als seine Ausschwingzeit, so dass sich dieses Filter durchaus gut für musikalische Anwendungen eignet.

## Vorschläge und Anmerkungen

Der Dynamikbereich kann durch die Verwendung einer höheren Versorgungsspannung weiter gesteigert werden. Wie bereits erwähnt, können LDRs normalerweise Signale bis zu 100 V verarbeiten. Um eine genauere und stabilere Frequenzregelung zu erreichen, könnte man fünf statt vier LDRs verwenden und einen in der Rückkopplungsschleife eines Operationsverstärkers einsetzen, der die LED ansteuert. Da das Filter mit Blick auf Musiksintesiser-Anwendungen entwickelt wurde, wurde die Verstärkung stark vom Q-Wert abhängig gemacht, um einen starken Anstieg des Ausgangssignals zu vermeiden. Alternativ könnte das Eingangssignal an den nichtinvertierenden Eingang von U1a statt an den invertierenden Eingang angelegt werden. Dies würde den Einfluss der Güte auf die Filterverstärkung

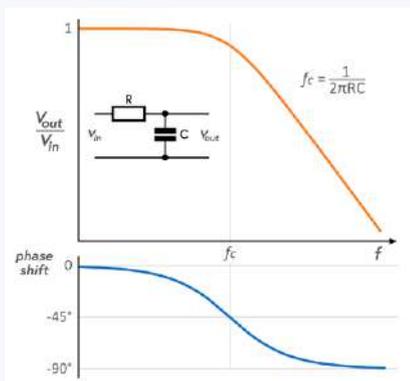
einschränken und das Filter würde eine nicht-invertierende Übertragungsfunktion erhalten. ◀

220331-02

## Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter [rs.elc.projects@gmail.com](mailto:rs.elc.projects@gmail.com) oder an Elektor unter [redaktion@elektor.de](mailto:redaktion@elektor.de).

## Steuerung der Filtergüte



Ein RC-Tiefpassfilter ist nicht nur ein Filter, sondern auch ein frequenzabhängiger Phasenschieber, der im Durchlassbereich bei  $0^\circ$  beginnt, bei der Grenzfrequenz auf  $-45^\circ$  geht und sich danach  $-90^\circ$  nähert. Werden vier dieser RC-Filter in Reihe geschaltet, beträgt die Phasenverschiebung bei der Grenzfrequenz  $4 \times -45^\circ = -180^\circ$ . Das bedeutet, dass das Signal bei dieser Frequenz das Vorzeichen wechselt. Wenn wir also eine negative Rückkopplung vom Ausgang zurück zum Eingang des Filters führen, hat dies eine positive Rückkopplung bei der Grenzfrequenz zur Folge, so dass eine Verstärkungsspitze bei dieser Frequenz erzeugt wird. Dadurch erhält das Filter den scharfen Übergang bei der Grenzfrequenz. Durch Einstellen des Gegenkopplungsanteils kann die Verstärkungsspitze variiert werden. Bei einer bestimmten maximalen Rückkopplung kann das Filter gar zu einem (Sinus-) Oszillator werden, der von 20 Hz bis 20 kHz abstimbar ist.

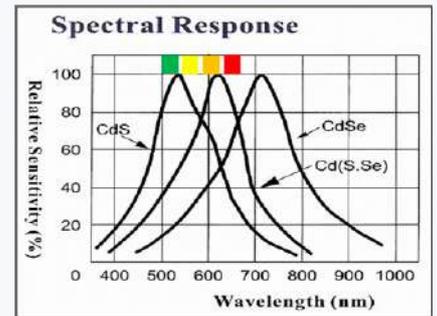
## Die LED-LDR-Kombination

### Aufbau

Die LED ist etwa 5...7 mm „hängend“ über den vier LDRs montiert. Die Anforderungen an die mechanische Präzision sind gering; die Messungen zeigten keinen signifikanten Unterschied im Ansprechverhalten, als die LED 1...2 mm aus der Achse verschoben wurde. Achten Sie aber darauf, die Schaltung vor Umgebungslicht abzuschirmen, da dieses auch das Filter moduliert!

### Wahl der LED-Farbe

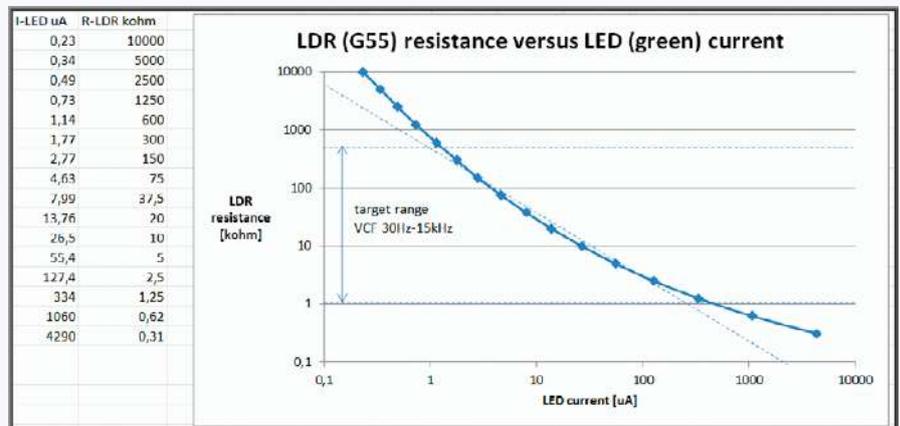
Ich habe LDRs auf Basis von Cadmium-Sulfid (CdS) verwendet, wobei eine grüne oder gelbe LED am besten funktioniert, wie im Diagramm des spektralen Ansprechverhaltens gezeigt.



Quelle: JCLH, Datenblatt des LDR GL5537-1

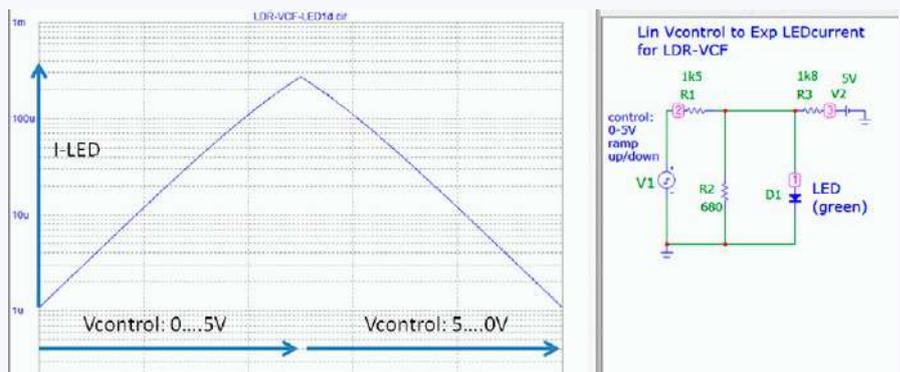
### Empfindlichkeit

Die Messungen ergaben, dass der LDR-Widerstand im Widerstandsbereich von 500 kΩ bis 1 kΩ nahezu eine lineare Funktion des LED-Stroms ist. Insgesamt konnte der LDR-Widerstand von 10 MΩ bis 300 Ω angesteuert werden, wie in diese Messkurve dargestellt.



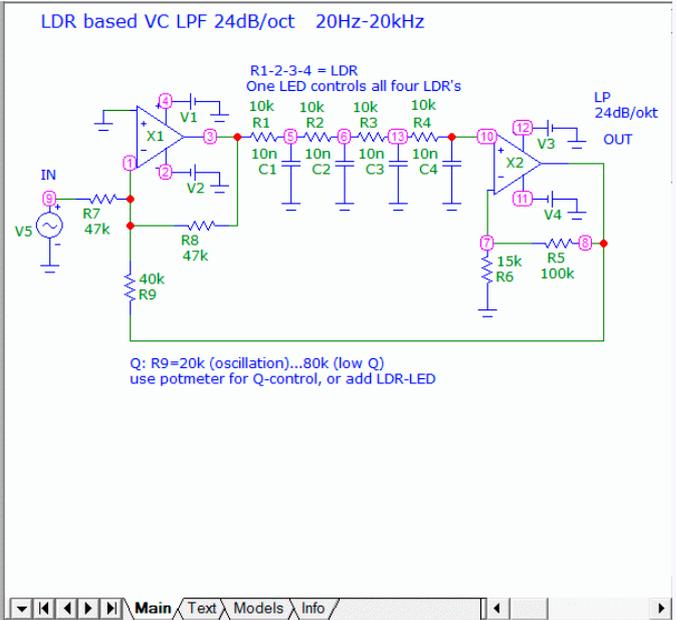
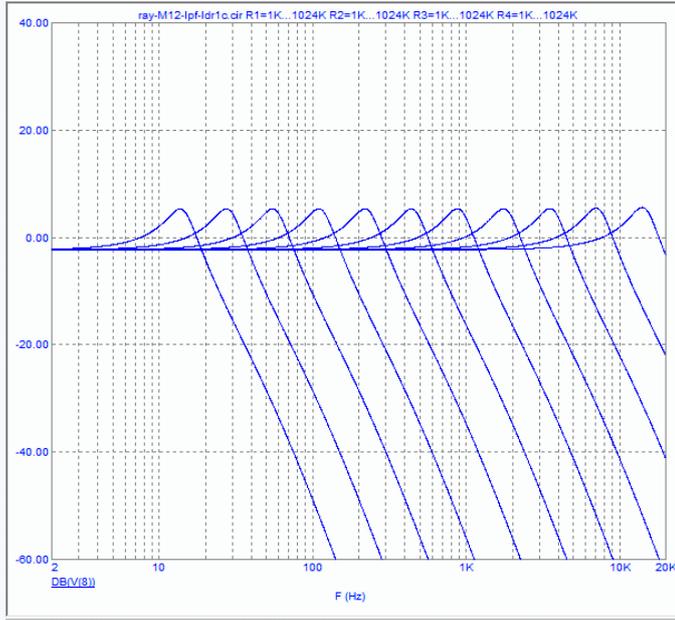
Hinweis: Hier wurde eine sehr helle grüne LED verwendet (OVLBG4C7). Schon 0,5 µA reichten aus, um sichtbares Licht zu erzeugen und den LDR-Widerstand zu beeinflussen.

Eine sehr einfache, aber ungenaue Methode zur Annäherung an eine exponentielle Steuerkurve ohne zusätzlichen Schaltungsaufwand wird hier gezeigt. Das Konzept ist: Wenn man eine Diode (hier die LED) mit einer Spannung steuert, ist der resultierende Strom eine Exponentialfunktion dieser Spannung. Es ist normalerweise keine gute Idee, eine Spannungsquelle direkt an eine LED anzuschließen, aber hier verwenden wir Widerstandswerte, die niedrig genug sind, um diesem Wert nahe zu kommen. Die Simulation unten zeigt, dass der LED-Strom exponentiell von 1 µA auf 300 µA ansteigt, wenn die Steuerspannung linear von 0 V auf 5 V angehoben wird. In der Praxis müssen Sie die Widerstandswerte möglicherweise ein wenig an Ihre LED anpassen. Wie bereits erwähnt, ist dies ziemlich ungenau und temperaturabhängig, aber der einfachste Weg, um eine exponentielle Steuerung für Synthesizer-Anwendungen zu erreichen. Die Klangbeispiele wurden mit dieser Steuerung erstellt.

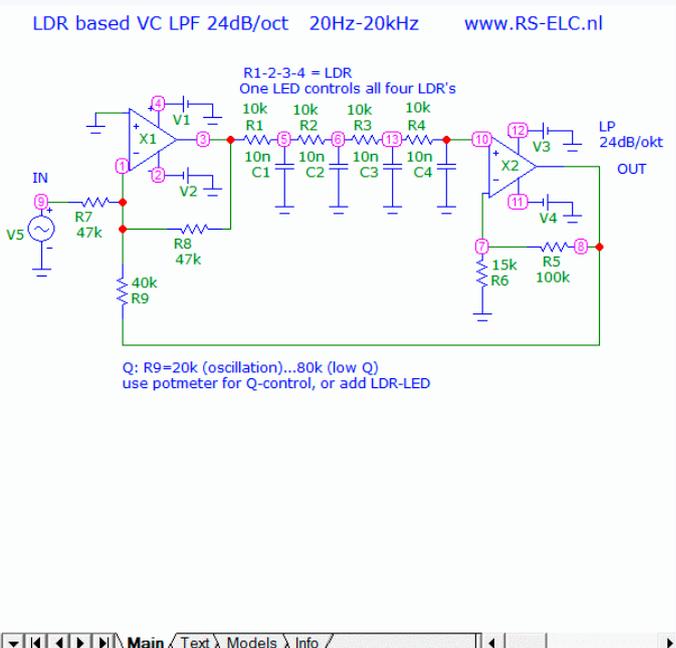
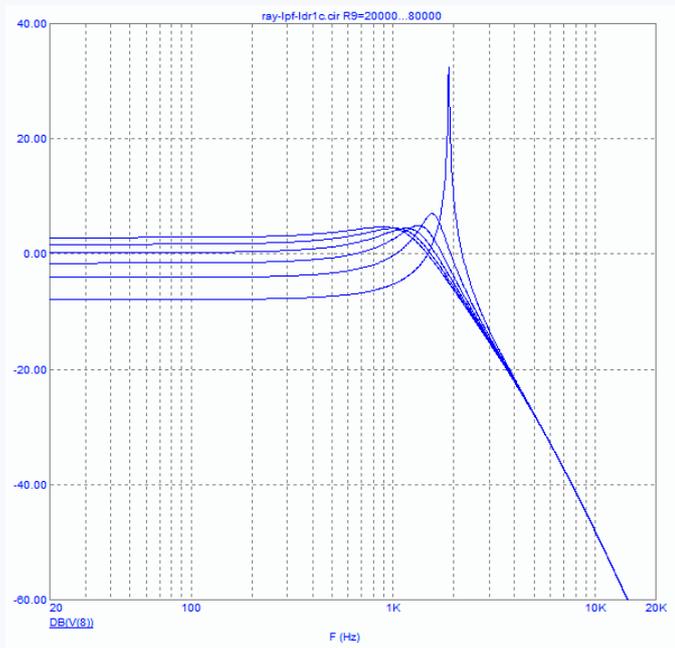




## Simulation der Filterantwort



In dieser Simulation wird die Grenzfrequenz des Filters von 20 Hz bis 20 kHz verschoben, wobei die Güte-Einstellung auf einem konstanten Wert gehalten wird.



Hier wird die Güte-Einstellung (P1) des Filters von 80 kΩ auf 20 kΩ verschoben und die Grenzfrequenz konstant gehalten.



### Passende Produkte

- > USB-Oszilloskop Joy-IT ScopeMega50 SKU 18277: [www.elektor.de/18277](http://www.elektor.de/18277)
- > MIDI-Eingangsplatine - Leerplatine SKU 18382: [www.elektor.de/18382](http://www.elektor.de/18382)
- > Gilles Brocard: The LTspice XVII Simulator  
Buch (Hardcover), SKU 19741: [www.elektor.de/19741](http://www.elektor.de/19741)  
E-Buch (PDF), SKU 20076: [www.elektor.de/20076](http://www.elektor.de/20076)

### WEBLINK

[1] Dieses Projekt bei Elektor Labs:  
<https://www.elektormagazine.de/labs/voltage-controlled-24dboct-synthesizer-filter-using-ldrs>

# Hochleistungsverstärker Gigant

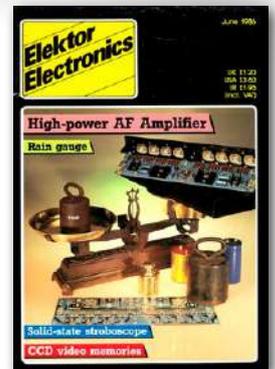
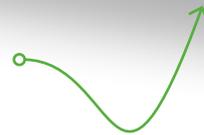
Der **Lauteste** von Allen!

Von Larry Kossek (Elektor)

Audio-Verstärker waren von Anfang an in der DNA von Elektor. Thomas Scherer baute sie alle, von 40 W bis 200 W, bevor er 1986 den Durchbruch mit einer Eigenkonstruktion versuchte, die einen hochwertigen Klang mit einer damals unglaublichen Ausgangsleistung von 1000 W liefern konnte.



Der Ausgangsteil ist ein Gegentakt-Design mit komplementären N- und P-Kanal-Leistungs-MOSFETs, die für gutes Einschwingverhalten und Linearität bei allen möglichen Ansteuerungspegeln ausgewählt wurden.



## Segway Cloners Club

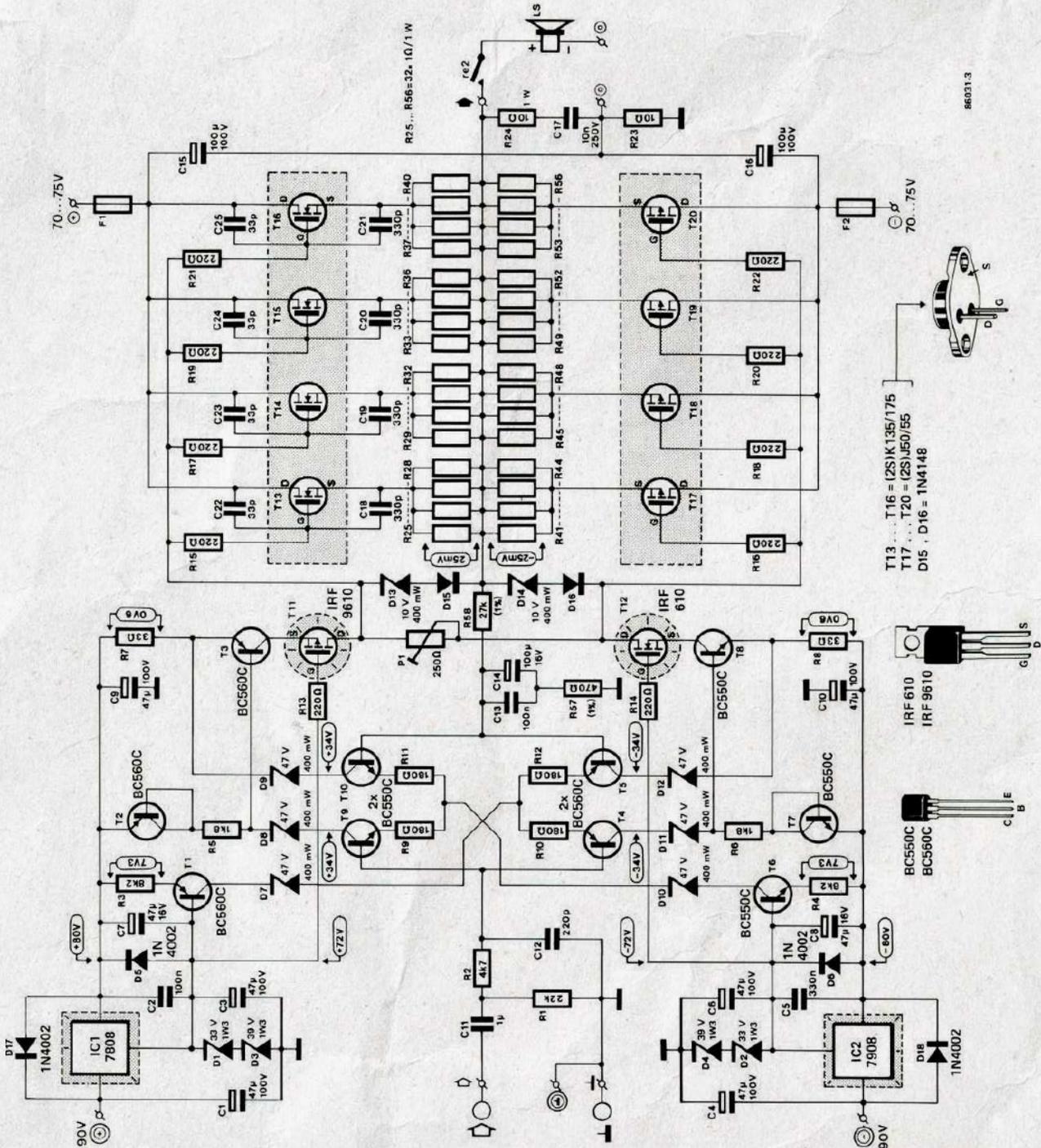
Ein Blick in die Elektor-Archive weckt Erinnerungen und fördert überraschende Assoziationen zutage. Thomas Scherer, der Mann, der 1986 diesen gigantischen NF-Verstärker entwarf, wurde später ein Fan und Förderer des Elektor Wheelie. Was haben ein 1000-Watt-Audioverstärker und ein Segway-Klon gemeinsam? Kleine Hilfe: Es ist ein Bauteil mit drei Beinen, einem Metallgehäuse und hoher Strombelastbarkeit. Richtig, es ist der MOSFET-Leistungstransistor, das Arbeitspferd, das die moderne Elektronik revolutioniert hat, wenn es um Leistung geht. Damals, in den achtziger Jahren, waren diese Leistungsbauteile in Elektor-Projekten noch nicht alltäglich. Und Thomas hatte schon damals eine Vorliebe für Hochstromschaltungen, die die Sicherungen des Labors zum Glühen brachten.

## „Wer Großes will, muss sich zusammenraffen“

Natürlich gehörten „normale“ Endstufenprojekte von Anfang an zur DNA von Elektor. Diese Verstärker kamen in rascher Folge und Thomas baute sie alle, von 40 W bis über 200 W. 1986 träumte er als Mitglied der Elektor-Redaktion von einem Audio-Verstärker, der so stark war, dass er mit der damals unvorstellbaren Leistung von 1000 W einen hochwertigen Klang liefern konnte. Er arbeitet hart und ging spät ins Bett, und viele seiner Träume enthielten komplementäre Ausgangstransistorpaare. „Ich entwarf die analoge Schaltung mit der Kraft des Kopfrechnens und mit ein wenig Hilfe eines programmierbaren Taschenrechners von Sharp. [...] Mein Vorschlag wurde von Elektor zur Begutachtung angenommen.“

## Eigenschaften

- > Ausgangsleistung: 2 × 250 W (8 Ω), 2 × 500 W (4 Ω), oder 1 × 1000 W (8 Ω)
- > Bandbreite: 8 Hz bis 100 kHz
- > Verzerrung: 0,1 % (bei 1 kW); 0,01 % (bei 600 W)
- > Dämpfungsfaktor: >100
- > Einschaltverzögerung, Lüftersteuerung, Spannungsüberwachung
- > Eingangspegel: 0,775 V<sub>eff</sub> bei voller Leistung
- > Gewicht: Das wollen Sie nicht wissen!



Beachten Sie die Symmetrie des Designs und die klare Trennung zwischen Spannungsverstärkung und Stromverstärkung.

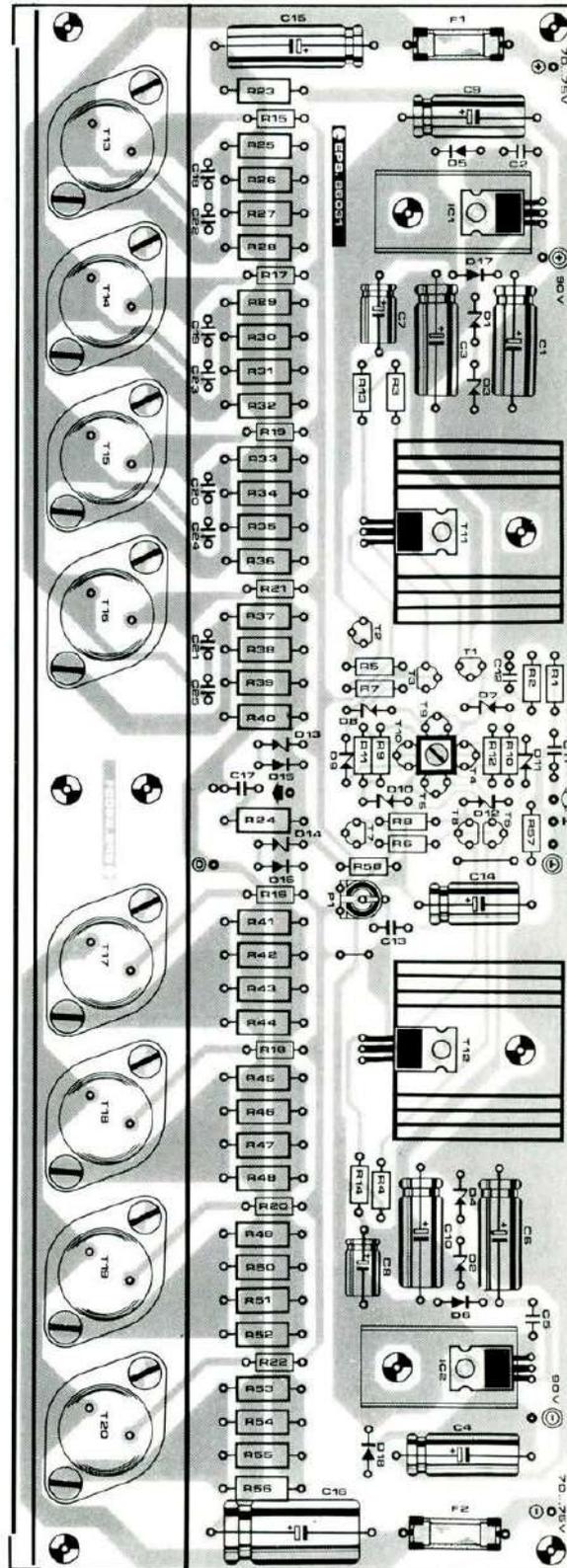
## Exorbitant teure MOSFETs

Bald musste Thomas vom theoretischen Schaltplan zu einem zuverlässig arbeitenden und reproduzierbaren Projekt übergehen: dem „Gigant“, ohne dabei zu viele Teile zu verheizen, insbesondere die exorbitant teuren Hitachi MOSFETs vom Typ 2SK135/2SK175 (T13-T16) und 2SJ50/2SJ55 (T17-T20).

„Bei Versorgungsspannungen von bis zu  $\pm 90$  V hatte ich mit Spitzenspannungen von über 125 V an den Lautsprecheranschlüssen zu kämpfen. Bis dahin war ich mit 48 V stets auf der sicheren Seite. [...] Es war wichtig, das Einschalten des Lautsprechers zu verzögern und einen Softstart für den Leistungstransformator zu realisieren. [...] Für Verzerrungsmessungen und einige andere Messungen diente mir der erschreckend teure Spektrumanalysator von Brüel & Kjær, der das Elektor-Audio-Labor zierte, zu dem ich nun Zugang hatte.“

Der Gigant-Verstärker (auch böswillig *Killerwatt* genannt) erregte großes Interesse und fand viele Jahre lang Anklang [1][2][3]. Auch mehr als 35 Jahre nach seiner Veröffentlichung taucht ab und an ein Exemplar des Kilowatt-Verstärkers im Internet zum Verkauf auf. Viele Leute haben zu Liedern, die durch dieses Ungetüm von einem Verstärker wirklich **LAUT** gespielt wurden, mit ihren funky Equipment geprotzt und möglicherweise die eine oder andere Sicherung durchgebrannt. ◀

220234-02



Die Schutzvorrichtungen zielen darauf ab, katastrophale Knallgeräusche in den Lautsprechern und durchgebrannte Netzsicherungen beim Einschalten des Verstärkers zu verhindern.



### Passendes Produkt

► **Elektor-Archiv 1974-2021 (USB-Stick) (SKU 20071)**  
[www.elektor.de/20071](http://www.elektor.de/20071)

## WEBLINKS

[1] T. Scherer, „Gigant (Teil 1)“, Elektor 5/1986: <https://www.elektormagazine.de/magazine/elektor-198605/48152>

[2] T. Scherer, „Gigant (2)“, Elektor 6/1986: <https://www.elektormagazine.de/magazine/elektor-198606/48164>

[3] T. Scherer, „Gigantomania“ (Retronik), Elektor 5/2016: <https://www.elektormagazine.de/magazine/elektor-201605/28928>



# Betreten für Unbefugte verboten!

Ein volumetrisches Display Made in Canada

Von Dan Foisy (Kanada) und Eric Bogers (Elektor)

Mit einer phasengesteuerten Anordnung von Ultraschallwandlern ist es möglich, einen kleinen Schaumstoffball nicht nur schweben zu lassen, sondern auch mit hoher Geschwindigkeit hin und her zu bewegen. In Kombination mit synchronisierten LED-Lampen lässt sich so dank des POV-Effekts (Persistence of Vision) ein „echtes“ 3D-Display zu bauen.

*Die Idee, einen kleinen Schaumstoffball auf einem Kissen aus Ultraschallwellen schweben zu lassen, ist eigentlich nicht neu. Mehr noch, diese Technik ist jetzt sogar für den durchschnittlichen Bastler verfügbar - siehe zum Beispiel den Artikel „Reiten auf der Schallwelle“ von Elektor-Redakteur Luc Lemmens, in dem er das Makerfab Acoustic Levitation Kit [1] diskutiert.*

*Die Inspiration für ein Projekt kann von überall her kommen, aber im Fall des Volumetric Display using an Acoustically Trapped Particle (vergessen Sie es, wir benutzen die handliche Abkürzung VDATP) war es ein Artikel von Forschern der Universität Sussex [2], in dem sie eine Methode beschrieben, ein durch Schallwellen eingefangenes Teilchen als eine Art Display zu verwenden. Als der in Toronto lebende Dan Foisy das dazugehörige Video [3] sah, war er völlig hin und weg. Angetrieben von seiner Erfahrung als Elektronikbastler machte er sich in seinem Heimlabor an die Arbeit. Lassen Sie es uns in Dans eigenen Worten hören.*

*„Mit Hilfe einer phasengesteuerten Anordnung von Ultraschallwandlern ist es möglich, eine kleine Schaumstoffkugel (1 mm) in der Schwebelage zu halten und sie schnell hin und her zu bewegen (mit mehr als 1 m/s). Unter Ausnutzung des Persistenz-Effekts ist es dann möglich mit diesem kleinen Ball Figuren in die Luft zu zeichnen, und zwar in einem Raumvolumen von etwa 100 × 100 × 140 mm. Mit Hilfe von RGB-LEDs kann die kleine Kugel auf ihrem Weg zu bestimmten Zeiten beleuchtet werden, was zu mehrfarbigen Bildern führt.“*

*Diese Art von Phased Array ist eigentlich nicht neu. Das Prinzip wird schon seit langem in umgekehrter Richtung etwa in der Radioastronomie angewendet, wo die Signale von vielen kleinen Antennen kombiniert werden, um ein sehr großes Radioteleskop zu simulieren (so groß, dass es in der Praxis nie gebaut werden könnte). Ein gutes Beispiel hierfür ist LOFAR [4][5], ein ursprünglich niederländisches Projekt, das sich inzwischen auf ganz*



Bild 1. Zwei Arrays aus jeweils 100 Ultraschallwandlern werden verwendet, um einen kleinen Schaumstoffball in der Luft schweben zu lassen.

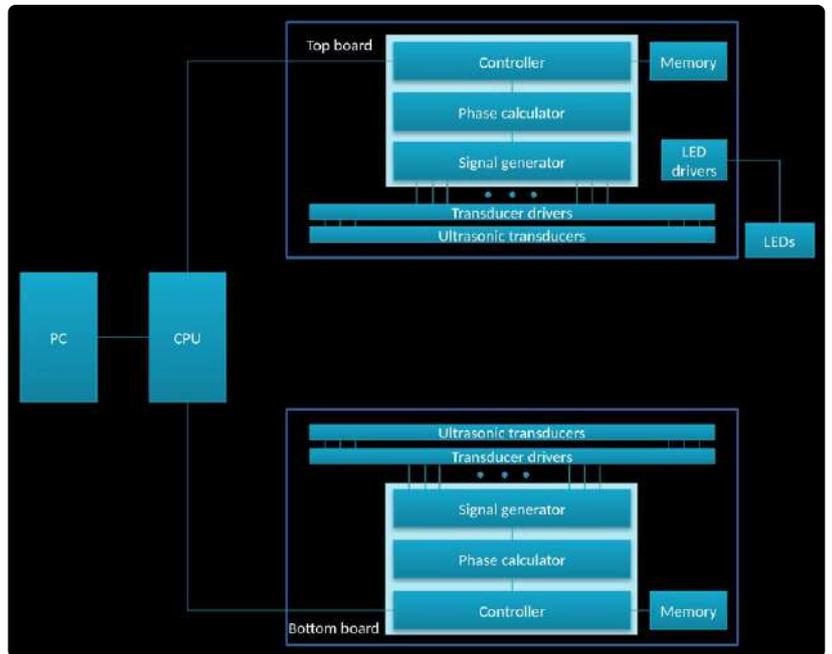


Bild 2. Blockschaltbild des VDATP. Abgesehen von den LED-Treibern sind die obere und untere Platine gleich.

Europa ausgedehnt hat und aus etwa 20.000 kleinen Einzelantennen besteht. Zusammen bilden sie eine phasengesteuerte Antennengruppe mit einer noch nie dagewesenen Auflösung von 0,2 Bogensekunden.

„Um den kleinen Ball nicht nur schweben zu lassen, sondern auch seine Position zu steuern, ist viel Rechenleistung erforderlich. Das VDATP, das ich gebaut habe, enthält neben der Haupt-CPU vier FPGAs, und es wandelt eine Position in die Phasensignale für die Ultraschallwandler um. Die FPGAs können die angezeigten Bilder drehen, verschieben und skalieren. Auch Animationen sind möglich; ich habe eine Animation eines Schmetterlings erstellt, der mit den Flügeln schlägt [6].“

Apropos US-Wandler: In diesem Projekt werden 200 Stück verwendet, die auf zwei Platinen verteilt sind. **Bild 1** gibt einen guten Eindruck von der Anordnung. Es sollte auch klar sein, dass dieses Projekt viel zu umfangreich ist, um es in Elektor ausführlich zu beschreiben. Wenn Sie sich also für das VDATP interessieren, sollten Sie Dan Foisy per E-Mail kontaktieren (siehe den Kasten Fragen oder Kommentare?).

„Das Blockdiagramm des VDATP war bereits eine Herausforderung für sich (siehe **Bild 2**). Es besteht hauptsächlich aus zwei Platinen, auf denen die US-Wandler untergebracht sind - eine oben und die andere unten. **Bild 3** zeigt eine der Platinen im Detail, **Bild 4** ein detaillierteres

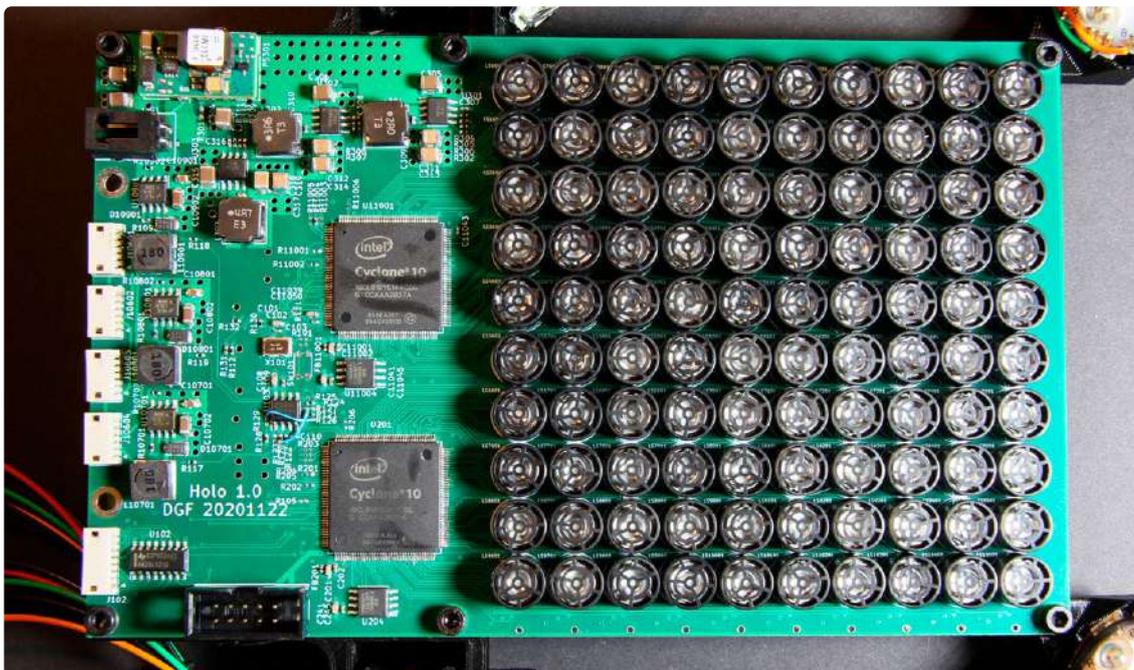
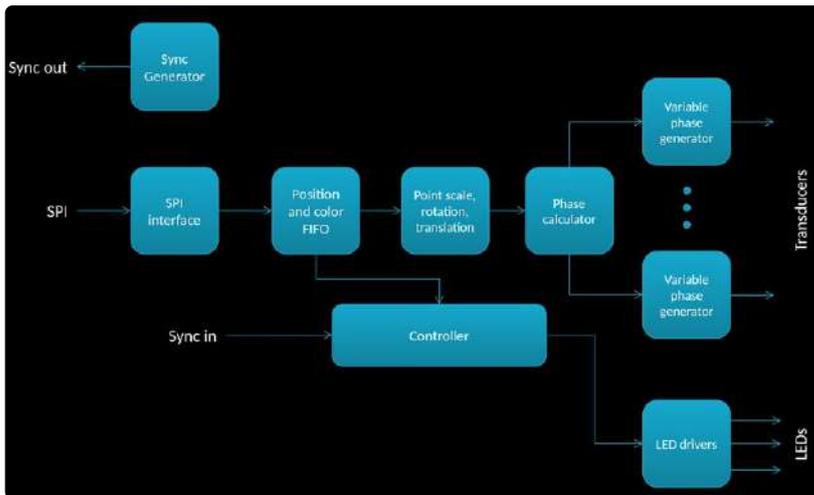


Bild 3. Eine der vollständig bestückten Leiterplatten. Jeder der 100 Wandler wurde vor dem Zusammenbau sorgfältig getestet.



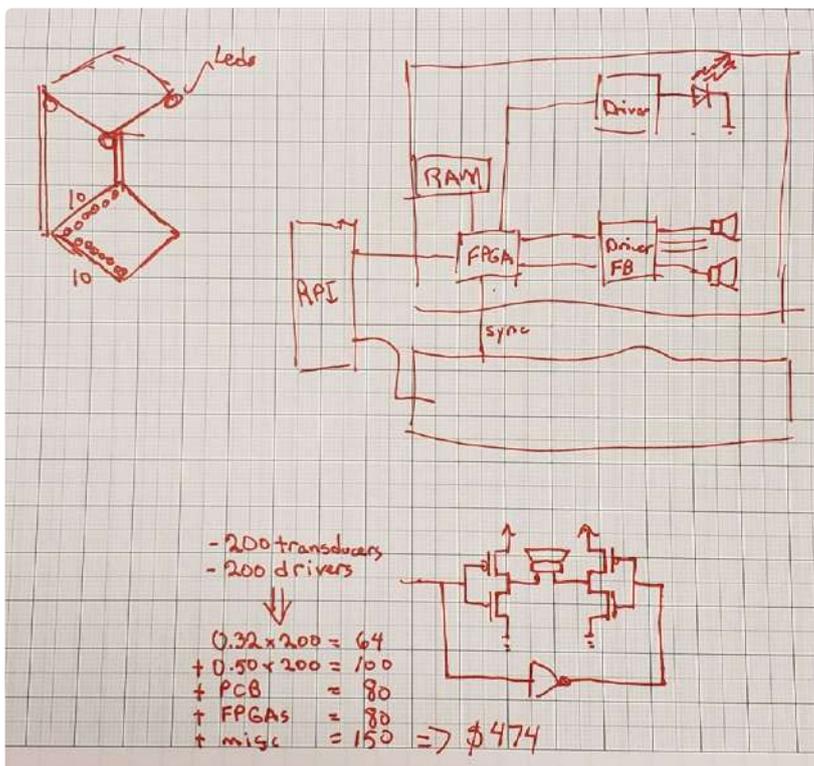
▲  
Bild 4. Das Blockdiagramm des FPGA-Codes.

Blockschaltbild der Platine, die auch für die Ansteuerung der LEDs zuständig ist.“

Wie jeder Maker weiß, ist der Weg von der ersten Idee bis zum funktionierenden Gerät lang und steinig, und das VDATP bildete in dieser Hinsicht keine Ausnahme. Bild 5 zeigt eine der ersten Skizzen und Bild 6 einen vollständig realisierten Teil des endgültigen Entwurfs. Wie in Bild 7 zu sehen ist, war KiCad für das Platinenlayout eine große, unverzichtbare Hilfe.

▼  
Bild 5. Eine der ersten Skizzen des VDATP.

„Es war ein sehr schönes Projekt, und es macht mir immer noch viel Freude, das VDATP in Aktion zu sehen. Es hat



etwas Magisches, ein Objekt in der Luft schweben zu lassen, das sich so schnell hin und her bewegt, dass man es nicht wirklich sehen kann - das gibt mir eine große Befriedigung.“ ◀

210237-02

### Über den Autor

Dan Foisy wurde Mitte der 1970er Jahre in Montreal, Kanada, geboren und wuchs in den Anfängen von Personal Computern und dem Internet auf. - eine wirklich aufregende Zeit. Ende der 1970er Jahre zog seine Familie nach Toronto.

Als er acht Jahre alt war, erhielt er einen 160-in-1-Elektronikbaukasten von Radio Shack (entspricht den bekannten Philips-Experimentierkästen in den Niederlanden), mit dem er Alarmer, Radios, Verstärker und andere Schaltungen baute. Er lernte auch, seine eigenen Schaltungen zu entwerfen und zu bauen. Später erwarb er ein Bachelor- und ein Master-Diplom in angewandten Wissenschaften an der Universität von Toronto. Während seines Masterstudiums arbeitete er am neu gegründeten Space Flight Lab am ersten kanadischen Weltraumteleskop mit dem Namen Microvariability and Oscillation of Stars (MOST)[7]. Danach war er für fünf weitere Satellitenprogramme verantwortlich.

Nach einer Reihe anderer Positionen arbeitet er jetzt bei der Royal Bank of Canada, wo er mehrere Teams leitet, die sich mit KI, insbesondere mit der Verarbeitung natürlicher Sprache, sowie mit der angemessenen und ethisch verantwortlichen Anwendung von KI befasst.

Als er seiner zukünftigen Frau den Heiratsantrag machte, schenkte er ihr einen traditionellen Diamantring als Verlobungsgeschenk. Seine Verlobte hielt es für angemessen, ihm im Gegenzug auch etwas zu schenken, und so kaufte sie eine ganze Reihe von Ausrüstung für sein Heimlabor - das er immer noch benutzt.

„Für mich war das ein wirklich toller Tausch :-)!“

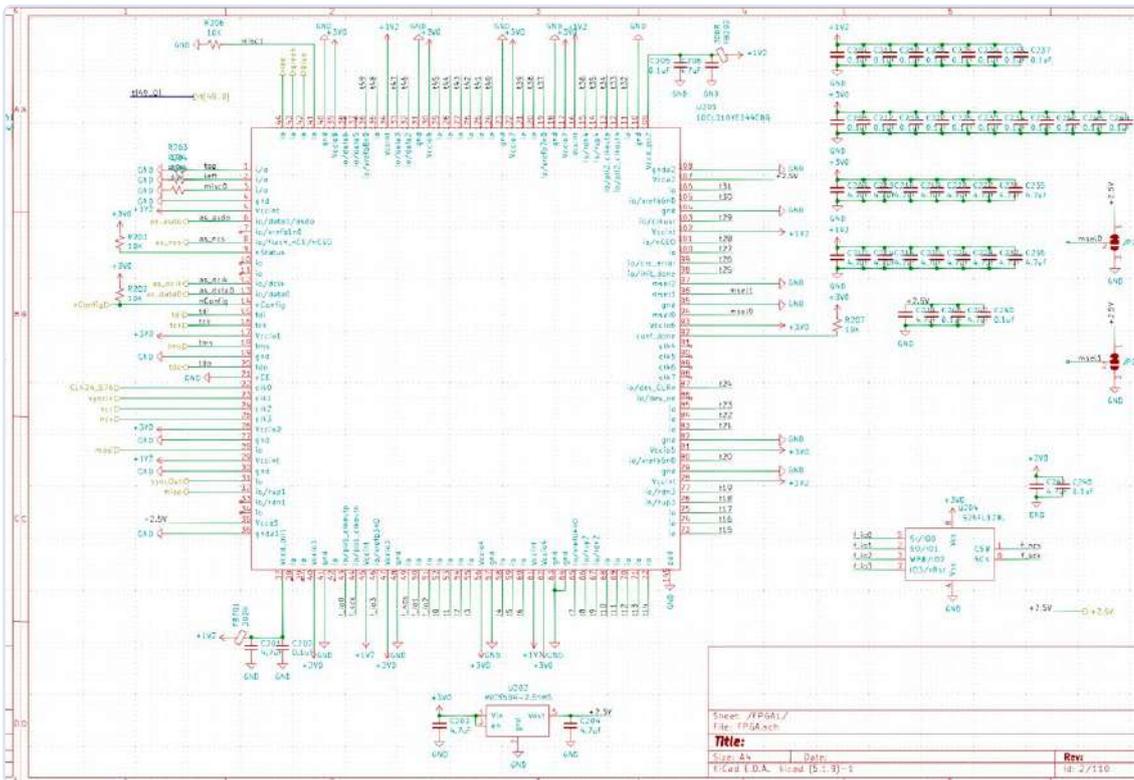
### Haben Sie Fragen oder Kommentare?

Wenn Sie daran interessiert sind, dies selbst zu bauen oder an weiteren Entwicklungen der volumetrischen Anzeige zu forschen, können Sie sich jederzeit an den Autor wenden: danfoisy@gmail.com.

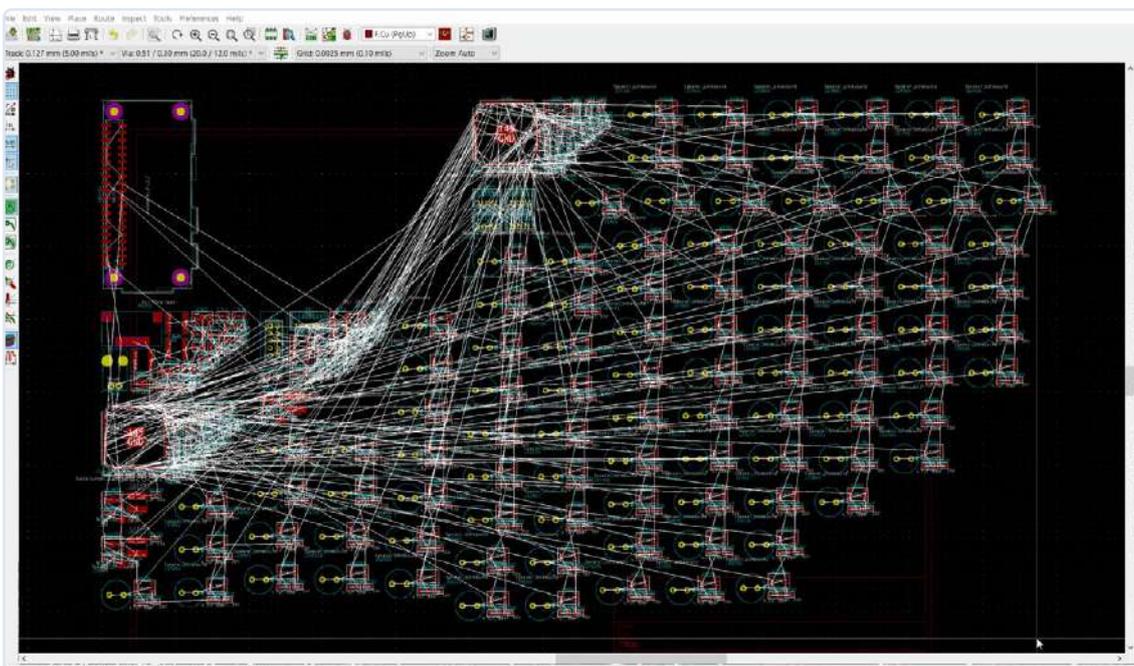


### Passendes Produkt

► **Makerfab Akustischer Levitator DIY-Kit (SKU 19984)**  
www.elektor.de/19984



◀ Bild 6. Die endgültige Version eines Teils der Schaltung. Eine Entwurfssoftware wie KiCad ist für Projekte dieses Umfangs praktisch unverzichtbar; ein Ansatz mit Papier und Bleistift ist einfach nicht gut genug.



◀ Bild 7. Dieses Rat's Nest zu entwirren war eine schwierige Aufgabe.

## WEBLINKS

- [1] Luc Lemmens, „Reiten auf der Schallwelle“, Elektor Juli/August 2022: <https://www.elektormagazine.de/magazine/elektor-265/60579>
- [2] Originalartikel der University of Sussex (2019): <https://sro.sussex.ac.uk/id/eprint/86930/>
- [3] VDTP-Video der University of Sussex: <https://youtu.be/Tm8JRI1q50>
- [4] LOFAR bei Astron: <https://www.astron.nl/telescopes/lofar/>
- [5] LOFAR (Wikipedia): <https://de.wikipedia.org/wiki/LOFAR>
- [6] VDTP-Video mit Schmetterlingsanimation von Dan Foisy: <https://youtu.be/hCC1C5KleUA>
- [7] MOST-Weltraumteleskop: <https://elektor.link/MOSTSpaceTelescope>

# Projekt 2.0

## Korrekturen, Updates und Leserbriefe

Redaktion: Jens Nickel

### Leistungshalbleiter-Tester

**Elektor Summer Circuits 2022, S. 56 (210707)**

Dieser Beitrag war für mich eine hervorragende Hilfe bei der Reparatur eines Schaltnetzteiles: Der „Verdächtige“ wurde ausgelötet, konnte somit eindeutig als fehlerhaft identifiziert und durch ein funktionierendes Bauteil aus einer anderen Platine ersetzt werden.

Mit Materialien aus dem Fundus war der Tester schnell zusammengebaut, aber eine passende 12-V-Lampe war nicht verfügbar. Daher kam ein vorhandener Widerstand mit  $47 \Omega / 5 \text{ W}$  (im Metallgehäuse) zum Einsatz. Der Spannungsabfall bringt nun (der unterschiedlichen Polarität wegen über einen Brückengleichrichter, Bauform DIL) eine LED (Vorwiderstand  $820 \Omega$ ) zum Leuchten (siehe Foto).

Danke an den Autor David Ashton!

Ulrich Strohmeyer



### High-End-Verstärker Fortissimo-100

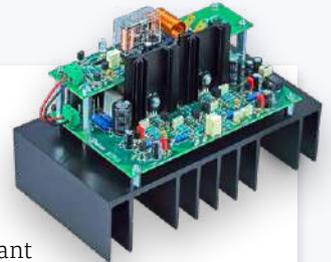
**Elektor 11-12/2022, S. 6 (210364)**

Ich finde den Verstärker interessant und der Artikel enthält eine Vielzahl von Daten. Aber ich vermisse Infos zum Stromverbrauch. Meine Frage lautet: Wie viel Watt verbraucht ein Kanal im Leerlauf und bei durchschnittlicher Leistung?

René Reynders

Im Ruhezustand beträgt die Gesamtstromaufnahme nach dem Aufwärmen maximal 350 mA, die Leistungsaufnahme liegt dann bei 28 W. Da die Stromversorgung eine stabile Spannung von  $\pm 40 \text{ V}$  aufweist, ist die maximale Durchschnittsleistung gleich der angegebenen Dauerleistung. Allerdings ist die durchschnittliche Ausgangsleistung bei der unverzerrten Ausgabe von Musik deutlich geringer als die angegebene kontinuierliche (maximale) Ausgangsleistung. Die Gesamtleistung eines Wechselstroms und einer Gleichspannung ist der Durchschnittswert des Stroms mal der Gleichspannung. Bei 181 W an  $4 \Omega$  beträgt der durchschnittliche Strom  $\sqrt{(181/2)/\pi} = 3,03 \text{ A}$ . Bei einer maximal unverzerrten Sinuswelle beträgt die Gesamtleistungsaufnahme an  $4 \Omega$  also  $2 \times 3,03 \text{ A} \times 40 \text{ V} = 242 \text{ W}$ . Ein Wirkungsgrad von 74,7%.

Ton Giesberts



### Nachruf

Mit großer Trauer geben wir bekannt, dass Peter Krengel, einer unserer Autoren, verstorben ist. Wir lernten Peter als kreativen „Maker“ kennen, der immer wieder neue Ideen hatte. Vom Löt-Reflow-Öfen bis zum Milliohmeter hatte er noch viele weitere Projekte in der Pipeline, ebenso wie Elektor-Bücher.

Peter, wir werden dich vermissen!

Das Elektor-Team

### Fliege-Bandsperre für Audio-Messungen

**Elektor 9-10/2022, S. 80 (210551)**

Der Artikel hat mich sofort begeistert - endlich ein sehr nützliches Analog-Projekt „alter Schule“ - mit aktuellen Bauteilen und hoher Qualität!

Ich habe mit dem Autor des Artikels Kontakt aufgenommen und muss das lobend erwähnen. Sehr freundlich und zuvorkommend! Einige Platinen hatte er übrig, und hat mir diese für einen attraktiven Preis unkompliziert zukommen lassen. Auch ergänzende Fragen und Varianten haben wir besprochen.

Ein guter Mix ist für diese tolle Zeitschrift das Entscheidende: Weiter so!

Jens Lemkamp

Vielen Dank für das Feedback und schön dass es so gut geklappt hat!

Jens Nickel, Redaktion

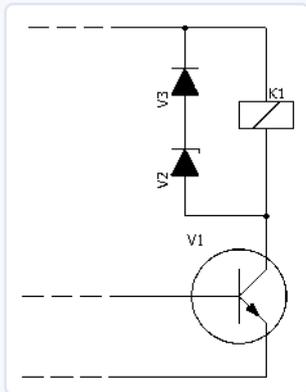


Bild 1

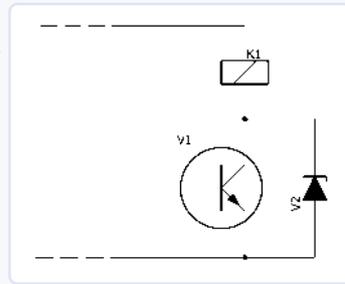


Bild 2

## Aller Anfang...

### Elektor 9-10/2022, S. 34 (220256)

Vielen Dank für Ihren Artikel im September/Oktober-Heft 2022, wo Sie sehr schön einige Anwendungen von Dioden zeigen, unter anderem in Bild 7 eine Anwendung als Freilaufdiode für ein Relais.

Ich habe solch eine Schaltung einmal verwendet, um ein Magnetventil in einer Abfüllanlage zu schalten. Dabei fiel mir auf, dass das Ventil schlagartig ein-, aber nur verzögert ausschaltete. Die Verzögerungszeit unterschied sich auch noch von Ventil zu Ventil, sodass ich sie leider nicht kompensieren konnte, zum Beispiel durch ein etwas früheres Abschalten.

Des Rätsels Lösung war natürlich, dass sich der Strom durch die Spule nur allmählich abbaute, weil die Freilaufdiode nur eine Durchlassspannung von 0,6 V aufwies, sodass irgendwann die Schließfeder die Oberhand gewann und das Ventil zaghaft gegen die restliche Magnetkraft schloss. Ich konnte das Problem durch den Einbau einer Z-Diode lösen, wie in Bild 1 gezeigt. Durch die Zenerspannung baut sich der Strom in der Spule erheblich schneller ab, was nicht nur die Ausschaltzeit deutlich verkürzt, sondern sich auch akustisch bemerkbar macht. Ein zartes „Pling“ wandelt sich in ein lautes „Dong“.

Zu meiner Überraschung habe ich diesen Kniff noch nie woanders gesehen, weder in der Literatur noch in realen Geräten.

Man kann die Schaltung noch etwas vereinfachen, indem man die eigentliche Freilaufdiode weglässt und die Z-Diode parallel zum Transistor schaltet (Bild 2).

Vielleicht eine interessante Variante auch für andere Elektor-Leser.

Thomas Klingbeil

## Was unser Bildungssystem von der Maker-Community lernen kann

Mit einem Alter von 34 Jahren befinde ich mich in der ungewöhnlichen Situation, in Deutschland eine Fachoberschule zur Erreichung der Fachhochschulreife (für ein anschließendes Studium) zu besuchen. Zeitgleich bin ich im Rahmen eines Programms zur Begabtenförderung an einer lokalen Universität für ein „Juniorstudium“ zugelassen. Dabei besuche ich parallel zur Schule Lehrveranstaltungen mit der Möglichkeit, schon einmal anrechnungsfähige Punkte für das Studium zu sammeln.

In der Schule hat sich im Prinzip seit über einer Dekade nichts verändert; vor der Klasse steht zumeist eine Lehrkraft und erzählt Unterrichtsstunde für Unterrichtsstunde theoretische Dinge, zu denen sich die Schüler nur wenig praktische Anwendungen vorstellen können. Die Lehrkräfte scheinen nicht nachvollziehen zu können, warum nach wochenlangem Unterricht zum Ohmschen Gesetz ein Versuch zur Ermittlung des Innenwiderstandes einer Spannungsquelle nur auf mäßiges Interesse stößt. Immer wieder kommt die Frage auf: „Wofür brauchen wir das eigentlich?“

Wäre die Maker-Community so aufgestellt, hätten wir ein Problem. Arduino und Co. zeichnet vor allem eines aus: Es werden sehr schnell Erfolgserlebnisse generiert. Umfangreiche Codebeispiele und verwandte Projekte bieten nahezu immer einen Startpunkt, mit dem man in kürzester Zeit seine Projektidee in die richtige Richtung bringen kann. Die bei Menschen generell gegebene Neugier nach dem „Warum?“ folgt spätestens, wenn Anpassungen nötig sind. Welcher Leser dieser Zeitschrift, mich eingeschlossen, hat nicht selbst anfangs Schaltungen quasi „blind“ nachgebaut? Würde man einen Projektartikel in Elektor lesen, wenn es parallel dazu keine fertige Schaltung gäbe? Oder diese erst in vier Wochen vorgestellt wird? Vermutlich eher nicht. Sicher kann man etwas so Gigantisches wie das Bildungssystem nicht schnell revolutionieren. Man könnte allerdings irgendwann einmal damit anfangen. Die Debatte über die „Jugend von heute“ ist mehrere Jahrzehnte alt. In meiner Jugend wurde die Schuld bei Computerspielen gesucht, heute sind vermeintlich die sozialen Medien schuld. Dabei unterstützen sowohl das Computerspiel- als auch das Social-Media-Argument meine These nur; beide vermitteln sehr schnell Erfolgserlebnisse. Man stelle sich vor, jedes Computerspiel oder jede soziale Plattform würde einen mehrwöchigen, theoretischen Vorbereitungskurs benötigen.

Ein sehr positives Beispiel bietet beispielsweise die Christian-Albrechts-Universität zu Kiel mit ihrem „Studieneingangsprojekt Elektrotechnik“ ([www.einfachgutelehre.uni-kiel.de/allgemein/studieneingangsprojekt-elektrotechnik](http://www.einfachgutelehre.uni-kiel.de/allgemein/studieneingangsprojekt-elektrotechnik)). Hier bauen Studierende im 1. Semester noch vor dem regulären Vorlesungsbeginn in Gruppen einen Metalldetektor zusammen. Das hierfür nötige Wissen wird in kurzen, klar verständlichen Vorträgen von den Dozenten vorgestellt. Jeder Vortrag vermittelt genau so viel Wissen, wie für die praktische Umsetzung nötig ist. Der Hauptbestandteil der Projektwoche war die Praxis, Verständnisfragen zum Hintergrund entstanden dabei von ganz alleine.

Sebastian Westerhold

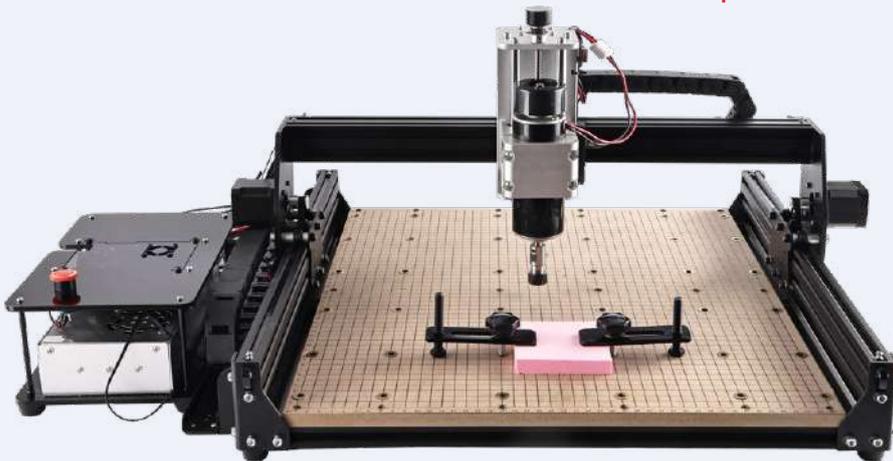
# Der Elektor Store

## Nie teuer, immer überraschend!

Der Elektor Store hat sich vom Community-Store für Elektor-eigene Produkte wie Bücher, Zeitschriften, Bausätze und Module zu einem umfassenden Webshop entwickelt, der einen großen Wert auf überraschende Elektronik legt.

Wir bieten die Produkte an, von denen wir selbst begeistert sind oder die wir einfach ausprobieren wollen. Wenn Sie einen Produktvorschlag haben, sind wir hier erreichbar ([sale@elektor.de](mailto:sale@elektor.de)).  
Unsere Bedingungen:  
**Nie teuer, immer überraschend!**

## Anet 4540 Desktop CNC-Fräsmaschine



**-200 €**

**699,00 €**

anstatt  
899,00 €

**Mitgliederpreis: 629,10 €**

 [www.elektor.de/20260](http://www.elektor.de/20260)

## Infrarot Reflow-Ofen T-962

Überarbeitete Elektor Version

**-40 €**

**229,00 €**

anstatt  
269,00 €

 [www.elektor.de/20346](http://www.elektor.de/20346)





## Great Scott Gadgets HackRF One SDR

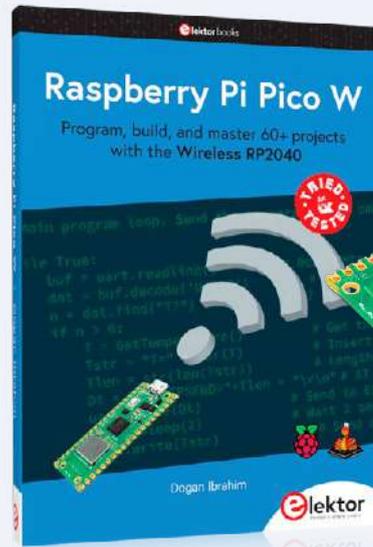
-40 €



299,00 €  
anstatt  
339,00 €

[www.elektor.de/18306](http://www.elektor.de/18306)

## Raspberry Pi Pico W



+  
GRATIS Pico W  
Board



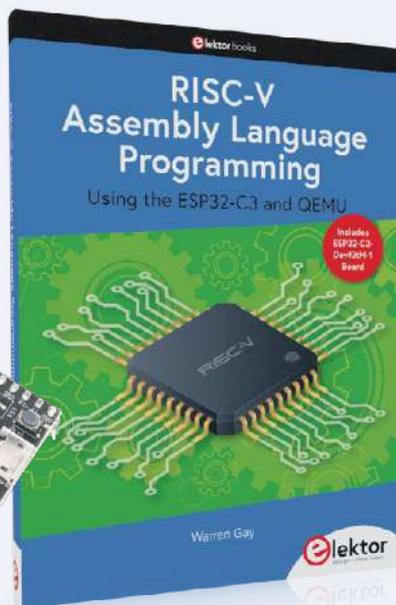
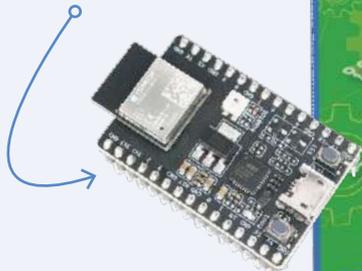
Preis: ~~47,90 €~~

Sonderpreis: 39,95 €

[www.elektor.de/20335](http://www.elektor.de/20335)

## RISC-V Assembly Language Programming

+  
GRATIS ESP32  
RISC-V Board



Preis: ~~59,90 €~~  
Sonderpreis: 39,95 €

[www.elektor.de/20296](http://www.elektor.de/20296)

## JOY-IT RD6006 Labornetzteil-Bundle



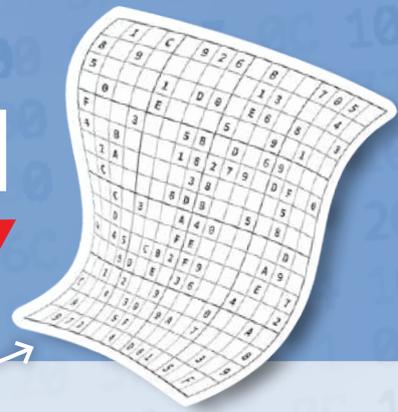
-35 €

Preis: ~~199,95 €~~  
Sonderpreis: 164,95 €

[www.elektor.de/19211](http://www.elektor.de/19211)

# Hexadoku

Sudoku für Elektroniker



Wie in jeder Ausgabe finden Sie auch in diesem Heft unser ganz spezielles Sudoku. PC, Oszilloskop und Lötkolben können sich erholen, während Ihre kleinen grauen Zellen auf Hochtouren arbeiten. Wenn Sie alle Hex-Ziffern in den grauen Kästchen herausgefunden haben, sollten Sie uns diese gleich zumailen – denn hier warten fünf Elektor-Gutscheine!

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist. Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass alle Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A bis F) in jeder Reihe, jeder Spalte und in jedem Fach mit 4 x 4 Kästchen (markiert durch die dickeren schwarzen Linien) **genau einmal** vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt. Wer das Rätsel löst – sprich die Zahlen in den grauen Kästchen herausfindet – kann einen von fünf Gutscheinen im Wert von 50 Euro gewinnen!



## EINSENDEN

Schicken Sie die Lösung (die Zahlen in den grauen Kästchen) per E-Mail, Fax oder Post an:

**Elektor Redaktion**  
Kackertstr. 10  
52072 Aachen

Fax: 0241 / 955 09-013

E-Mail: [hexadoku@elektor.de](mailto:hexadoku@elektor.de)

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der  
15. Februar 2023.

## DIE GEWINNER DES HEXADOKUS AUS DER AUSGABE NOVEMBER/DEZEMBER STEHEN FEST!

Die richtige Lösung ist: **DBE39**

Aus allen Einsendungen mit der richtigen Lösung haben wir die fünf Gewinner eines Elektor-Wertgutscheins über je 50 € gezogen.

Die Namen der Gewinner werden unter [www.elektormagazine.de/hexadoku](http://www.elektormagazine.de/hexadoku) bekannt gegeben.

Herzlichen Glückwunsch!

E				B 8			3 F								1
	3			6 F		7 D		1 A							8
			7			A B					2				
	D 8			1 3 5	C 7	E				A F					
	8 F 3			4		B				9 D C					
B				2		9 8		A							7
	A E		8		D 1 7	F		4		B 0					
	0 9		B						D		2 5				
	9 7		E						C		3 A				
	E 5		7		A 2 F	1		9		4 6					
D				3		B 4		6							5
	6 A 1			F			2			8 9 B					
	C 3			5 B E	9 D	4				6 2					
			0			6 A				3					
	B			F 4		3 2		8 1						D	
2				9 C			6 0								B

7	B	8	C	2	A	6	D	1	4	5	E	F	0	9	3
9	D	E	5	1	0	3	8	A	F	6	B	2	7	4	C
F	0	2	3	4	B	E	7	9	C	D	8	5	6	1	A
A	6	4	1	9	C	5	F	7	0	2	3	B	8	D	E
4	7	9	8	A	6	B	1	5	D	E	0	3	C	F	2
B	1	F	A	C	3	0	E	2	6	4	7	D	9	5	8
C	2	D	6	5	7	8	9	B	1	3	F	E	4	A	0
E	3	5	0	D	F	2	4	8	9	A	C	1	B	6	7
D	8	B	9	7	4	C	5	F	2	0	1	A	E	3	6
2	A	7	F	6	8	D	B	E	3	9	4	0	5	C	1
1	C	0	E	F	9	A	3	6	7	8	5	4	D	2	B
6	5	3	4	E	2	1	0	C	A	B	D	7	F	8	9
5	9	1	2	8	E	F	6	D	B	7	A	C	3	0	4
0	E	A	B	3	D	9	C	4	8	F	2	6	1	7	5
8	4	C	D	0	5	7	2	3	E	1	6	9	A	B	F
3	F	6	7	B	1	4	A	0	5	C	9	8	2	E	D

Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.

Von Arduino und Elektor für Elektronik-Fans erstellt

# Jetzt Erhältlich

Ein großartiges Elektor Magazine erstellt zusammen mit Arduino!

DIY-Elektronikprojekte, technische Inhalte und vieles mehr von Arduino und Elektor

Viele tolle Projekte und Tutorials

Steigt ein in Inhalte wie MicroPython, TinyML und Heimautomatisierung mit Arduino

Lernt die Arduino Macher kennen: Fabio, Massimo und David

Lernt Alles über das Portenta X8

Ab jetzt in Kiosken, Onlineshops und natürlich beim Elektor Store erhältlich!

In allen Artikeln sind Links zu passenden Lösungen von Arduino gegeben. Somit wird es noch leichter an weitere Infos zu kommen.



Jetzt Kaufen!

[www.elektor.de/arduino-magazine](http://www.elektor.de/arduino-magazine)



