NOVEMBER/DE7EMBER 2021





Erkennen von Objekten mit Nvidia Jetson Nano und Edge Impulse

S. 6

## In dieser Ausgabe

- > Elektor wird 60: Es werde Licht!
- > Blitzlicht-Analogspeicher
- > Bauen Sie Ihren Hochpräzisionskalibrator
- > Arduino Nano RP2040 Connect
- > CO<sub>2</sub>-Messgerät-Set fürs Klassenzimmer
- > Bluetooth-Beacons in der Praxis
- > GUIs mit Python und guizero
- > Neuronen in neuronalen Netzen verstehen: Logische Neuronen
- HomeLab: Die Wiedergeburt des Junior-Computers
- > Nostalgisches MW/LW-Radio
- > Poster LCR-Meter

## und vieles mehr!

Open-Source-GPS-Tracking-Plattform Fahrzeug-Tracking ohne Cloud









EMV-Vorkonformitätstest mit LISN Für Ihr DC-versorgtes Projekt





## **UNSERE SORTIMENT** VON TECHNIKERN FÜR TECHNIKER

## The best part of your project: www.reichelt.de/sortiment

## Nur das Beste für Sie – von über 900 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.





über 120.000 ausgesuchte Produkte

zuverlässige Lieferung – aus Deutschland in alle Welt

elektronik - The best part of your project

reich

## www.reichelt.de Bestellhotline: +49 (0)4422 955-333

und Produktionsmaterialien

finden Sie online!

**Gleich entdecken** www.reichelt.de/3d-druck

Top Preis-Leistungs-Verhältnis

Es gelten die gesetzlichen Widerrufsregelungen. Alle angegebenen Preise in € inklusive der gesetzlichen MwSt., zzgl. Versandspesen für den αesamten Warenkorb. Es gelten ausschließlich unsere AGB (unter 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

## **IMPRESSUM**

52. Jahrgang, Nr. 582 November/Dezember 2021 ISSN 0932-5468

Erscheinungsweise: 9x jährlich (6x Elektor-Doppelheft + 3x Elektor Industry Magazin)

### Verlag

Elektor Verlag GmbH Kackertstraße 10 52072 Aachen Tel. 0241 95509190

Technische Fragen bitten wir per E-Mail an redaktion@elektor.de zu richten.

## Hauptsitz des Verlags

Elektor International Media Postbus 11, 6114 ZG Susteren Niederlande

### Anzeigen

Raoul Morreau (Leitung) Mobil: +31 6 440 399 07 E-Mail: raoul.morreau@elektor.com

**Büsra Kas** Tel. 0241 95509178 E-Mail: busra.kas@elektor.com

Es gilt die Anzeigenpreisliste ab 01.01.2021.

### Distribution

**IPS Pressevertrieb GmbH** Postfach 12 11, 53334 Meckenheim Tel. 02225 88010 Fax 02225 8801199

### Druck

Pijper Media, Groningen (NL)

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.

EDITORIAL

## von Jens Nickel

Chefredakteur ElektorMag



## Machine Learning leicht gemacht

Es gibt im Moment wohl kaum ein faszinierenderes Thema in der IT/Elektronik als das Maschinelle Lernen durch Neuronale Netze. Anwendungen wie die Sprachsteuerung und Bilderkennung laufen auch auf kompakten, energiesparenden und preiswerten Controllerboards. Grundlage hierfür ist, dass das Training des Netzes bereits auf leistungsstarken Plattformen durchgeführt wurde, zum Beispiel in der Cloud. Das Aufnehmen und Aufbereiten der nötigen Daten, das Trainieren und Testen des Netzes und schließlich die Übertragung auf den ausführenden Mikrocontroller ist aber nicht einfach - gerade für die große Schar von Elektronikern, die (endlich) einen Einstieg in diesen spannenden Bereich finden möchte, sei es rein zur persönlichen Fortbildung oder den Beruf.

Hier kommen ML-Frameworks wie Edge Impulse ins Spiel. Für einen ersten Test ist nicht einmal eines der unterstützten Controllerboards nötig. Ein Smartphone genügt, um Bilder oder Töne aufzunehmen und ein erstes Netz so zu trainieren, dass Muster erkannt werden. Mein Kollege Mathias Claußen hat das Ganze mit einem Nvidia Jetson Nano, einer einfachen Kamera und einem Stapel Spielkarten ausprobiert (Seite 6). Sein Artikel ist eine gute Schrittfür-Schritt-Anleitung für den Einstieg und lädt zum Mitmachen ein!

Herzlich einladen möchten wir Sie auch auf die productronica, die vom 16. bis zum 19. November in der bayerischen Landeshauptstadt stattfindet. Gemeinsam mit der Messe München richten wir auch in diesem Jahr den bekannten "Fast-Forward"-Startup-Wettbewerb aus (www. elektormagazine.com/pffwd21).

Doch das ist noch nicht alles, denn wir wollen sechs Jahrzehnte Elektor auch auf der productronica entsprechend würdigen. In diesem Jahr veranstalten wir zusammen mit der Messe und der Zeitschrift Elektronikpraxis das World Ethical Electronics Forum, kurz WEEF (www. elektormagazine.com/weef). Zum Thema Ethik und Nachhaltigkeit in der Elektronik erwarten Sie interessante Vorträge, moderiert von meinem Kollegen Stuart Cording. Seien Sie dabei!

## **ELEKTOR WIRD 60: ES KOMMT NOCH MEHR!**

Elektor feiert 60 Jahre Verlagsgeschichte und Elektronikinnovation mit einigen spannenden Sonderprojekten. Wir arbeiten an einer Sonderausgabe von Elektor-Industry (November 2021) und dem World Ethical Electronics Forum 2021, das am 18. November 2021 stattfinden wird. Haben Sie Ideen für weitere Sonderprojekte? Schicken Sie mir Ihre Ideen: shenja.panik@elektor.com!



Shenja Panik, "Elektor 60" Festkomitee-Koordinatorin

## - Unser Team

Chefredakteur: Redaktion:

Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf, Dr. Thomas Scherer, Clemens Valens Leserservice: Ralf Schmiedel Elektor-Labor: Mathias Claußen, Ton Giesberts, Luc Lemmens, Clemens Valens Grafik & Layout: Giel Dols, Harmen Heida Herausgeber: Erik Jansen

Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)

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

FIPP

FACHPRESSE

DEUTSCHE

Elektor ist Mitglied von FIPP, einer Organisation, die "über fast 100 Jahre gewachsen ist und Medienbesitzer und Content-Ersteller aus der ganzen Welt umfasst".





## Bildverarbeitung für Einsteiger

Erkennen von Objekten mit dem Nvidia Jetson Nano und Edge Impulse

## Rubriken

- 3 Impressum
- 14 Elektor-Jumpstarter-News Kommende Kampagnen
- **31** Aller Anfang ... ist gar nicht schwer! Weiter geht's mit der Spule...
- 80 Betreten verboten! Wie der Junior-Computer wieder zum Leben erweckt wurde
- 92 Elektor Ethics Der physische Körper der Künstlichen Intelligenz
- 94 Projekt 2.0 Korrekturen, Updates und Leserbriefe
- **114 Hexadoku** Sudoku für Elektroniker



## Hintergrund



- 22 Multifunktions-Komponententester LCR-T7 von JOY-iT Testen von passiven Bauteilen, diskreten Halbleitern und IR-Fernbedienungen
- **34** Neuronen in neuronalen Netzen verstehen Teil 2: Logische Neuronen
- 50 Bluetooth-Beacons in der Praxis Ortsbestimmung in Innenräumen
- 57 C-Programmierung auf dem Raspberry Pi Ein Beispielkapitel: Kommunizieren über WLAN
- 69 Parallax Propeller 2 Teil 5: Das Innenleben des Smart Pins
- 89 Arduino Nano RP2040 Connect Raspberry Pi RP2040 + WLAN + Bluetooth
- 96 Grafische Benutzeroberflächen mit Python und guizero Einführung in guizero
- 108 Elektor wird 60 Es werde Licht!



**CO<sub>2</sub>-Messgerät fürs Klassenzimmer** Ein ESP8266-basiertes Gerät

101



## Projekte

- 6 Nvidia Jetson Nano Bildverarbeitung für Einsteiger Teil 2: Erkennen von Objekten mit Edge Impulse
- 16 Open-Source-GPS-Tracking-Plattform Fahrzeug-Tracking ohne fremde Cloud mit Traccar
- 24 Rausch-Synthesizer Vom Rauschen zur Musik mit dem PRBSynth1
- 40 Probleme mit der Sicherheit? Bekämpfen Sie Feuer mit Feuer! Blitzlicht-Analogspeicher-Zusatz für das manipulationssichere Datenpaket
- **48 Neues LCR-Meter 50 Hz 2 MHz** Das Jumpstarter-Projekt im Überblick!
- 60 EMV-Vorkonformitätstest für Ihr DC-versorgtes Projekt Teil 2: Die Hardware und wie man sie benutzt
- 74 Modbus über WLAN Teil 1: Hardware und Programmierung
- 84 Bauen Sie Ihren eigenen Hochpräzisionskalibrator -10 V bis +10 V, 0 bis 40 mA, 0,001%
- 101 CO<sub>2</sub>-Messgerät-Set fürs Klassenzimmer Ein ESP8266-basiertes Gerät, entwickelt von der FH Aachen
- **106** Nostalgisches MW/LW-Radio mit MK484 ...macht immer Spaß zu bauen!

## Vorschau

## Elektor Januar/Februar 2022

Das nächste Heft ist wie immer randvoll gefüllt mit Schaltungsprojekten, Grundlagen sowie Tipps und Tricks für Elektroniker.

## Aus dem Inhalt:

- > Pi-KVM: Fernsteuerung anderer Computer
- > Preiswertes Robotik-Kit und Motorsteuerung
- > Audio-Messungen mit dem USB-Audio-Interface
- > Vielseitiger Servotester
- > Reparatur von Lithium-Batteriepacks
- > Over-the-Air: Updates für Arduino und ESP
- > Mittelgroße Elektromotoren
- > NB-IoT in der Praxis

## Und vieles mehr!

Elektor Januar/Februar 2022 erscheint am 6. Januar 2022. Änderungen vorbehalten.







# Nvidia Jetson Nano



Bild 1. Jetson Nano Developer Kit (Quelle: developer.nvidia.com).

Teil 2: Erkennen von Objekten mit Edge Impulse

### Von Mathias Claußen (Elektor-Labor)

Wer eine einfache Lösung für den Start in die Welt neuronaler Netzwerke sucht, sollte sich Edge Impulse anschauen. Daten wie zum Beispiel Kamerabilder lassen sich mit den unterschiedlichsten Boards und Geräten erfassen. um in der Cloud neuronale Netze zu erstellen. Diese können dann in eigenen Anwendungen zum Einsatz kommen. Wir haben das Ganze mit einem Jetson Nano und einem Kartenspiel ausprobiert.

Nach der Vorstellung des Jetson Nano (Bild 1) und dem SparkFun JetBot AI Kit (Bild 2) im ersten Teil der Serie [1] wird es nun Zeit, die ersten Schritte mit der Bilderkennung zu machen. Um Objekte mit dem Jetson Nano zu erkennen, bedarf es Software und eines neuronalen Netzes, das auf die Objekte trainiert wurde. Wie schon im Online-Artikel "Kaffee zubereiten mit MAX78000 und einer Prise KI" [2] beschrieben, ist das Training dieser Netzwerke ein nicht ganz einfaches Unterfangen. PyTorch oder auch TensorFlow kommen im ersten Schritt als Konsolenanwendung daher, in deren Bedienung man sich erst einarbeiten muss. Auch benötigt das Erstellen eines neuronalen Netzwerkes auf dem eigenen Rechner je nach Grafikkarte und CPU einige Zeit.

Doch damit hören die Herausforderungen nicht auf: Ein neuronales Netzwerk muss zuerst mit einem Datensatz trainiert werden. Um die Daten zu sammeln und für das Training aufzubereiten, sind weitere Anwendungen nötig. Daher kann es einige Zeit dauern, bis das erste neuronale Netz trainiert ist.

## Edge Impulse

Wer eine einfachere Lösung für den Start in die Welt neuronaler Netzwerke sucht, sollte sich Edge Impulse [3] anschauen. Edge Impulse ist ein webbasierte Plattform, um mit eigenen Daten passende neuronale Netzwerke zu erstellen und auf unterschiedlichste Geräte zu bringen. Dabei reicht die Palette von einem Arduino bis hin zu einem

x86/AMD64-basierenden Laptop. In diesem Artikel nutzen wir Edge Impulse, um ein neuronales Netzwerk zu erstellen, mit dem der JetBot ein paar Objekte erkennen soll.

Daten lassen sich mit unterschiedlichen Controllerboards vom PC aus mit einem Webbrowser oder auch dem Smartphone erfassen. Die Daten werden direkt an die Server von Edge Impulse gesendet und auch dort gespeichert. Wer schon Daten gesammelt hat, kann diese einfach über die Weboberfläche hochladen.

Das Sortieren und Beschriften der Daten geschieht dann über die Weboberfläche. Dort werden auch die Parameter zum Erstellen des neuronalen Netzes festgelegt und vorgeschlagen.

Das Erstellen eines neuronalen Netzes ist sehr rechenintensiv. Daher werden die Daten, die auf die Server von Edge Impulse hochgeladen wurden, dort auch in ein neuronales Netz umgesetzt. Das Training passiert also in der Cloud und belastet nicht den eigenen Rechner. Das erstellte neuronale Netz kann dann passend aufbereitet als Programmbibliothek heruntergeladen und in eigenen Anwendungen genutzt werden.

All dies werden wir nachfolgend anhand einer kleinen Bilderkennungsanwendung vorführen. In unserer Schwesterzeitschrift Elektor Industry ist darüber hinaus ein Artikel erschienen, der eine Geräuscherkennung mit Edge Impulse demonstriert [4].



Bild 2. SparkFun JetBot AI Kit (Quelle: SparkFun).



Bild 3. UNO-Karten.





## **Eine Runde Karten**

Als Testobjekte wollten wir hier Dinge nutzen, die ein Mensch recht gut auseinanderhalten kann. Die Wahl fiel auf ein paar Karten aus dem Uno<sup>®</sup> Kartenspiel von Mattel (**Bild 3**). Die Karten sind hier exemplarisch gewählt und könnten auch Verkehrsschilder sein, jedoch war ein Satz Uno-Karten gerade zur Hand. Als Plattform wird ein Jetson Nano plus ein SparkFun JetBot Al Kit (Bild 2) verwendet, das eine 8-Megapixel-Kamera enthält.

Das erste Ziel soll sein, die Uno-Karte Eins, die Karte Zwei und die Rückseite der Karten auseinanderhalten zu können. Dazu muss wie schon erwähnt ein neuronales Netz trainiert werden, am besten mit Bildern der zu erkennenden Objekte. Daher wollen wir uns nun erst einmal mit der Datenerfassung und der Oberfläche von Edge Impulse beschäftigen.

## **Edge Impulse Projekt**

Für die Verwendung von Edge Impulse ist eine Anmeldung erforderlich. Für dieses Projekt nutzen wir die kostenfreie Variante des Dienstes, die ein paar Einschränkungen aufweist (dazu später mehr). Nach der Anmeldung kann ein neues Projekt erstellt werden; ein Wizard (**Bild 4**) begrüßt uns.

Auf die Frage, welche Art von Daten wir verarbeiten möchten, antworten wir mit *Images*, da Bilder der Uno-Karten verarbeitet werden sollen. Für die Uno-Karten soll in diesem Fall eine Klassifizierung (*Classify a single object*) erfolgen, denn wir wollen im ersten Schritt nur eine Karte verarbeiten und bestimmen, um welche es sich handelt, wie **Bild 5** zeigt. Damit ist das Projekt mit den Parametern für die Erkennung bereit. Was nun kommt ist das Sammeln von Daten, mit denen das Training beginnen kann.

## Datenerfassung

Die Daten zum Trainieren des neuronalen Netzwerkes generieren wir in diesem Projekt selbst. Für die Bilderkennung gibt es auch Datensätze mit tausenden Bildern, die schon in gewisse Gruppen vorsortiert sind; je nach Anwendungsfall kann es sinnvoll sein, auf einen solchen Datensatz zurückzugreifen. Da Uno-Karten dort nicht direkt zu finden sind, müssen wir aber unseren eigenen Datensatz generieren. Dies soll auch demonstrieren, wie man selbst Bilder zum Anlernen erstellt und was eventuell zu nicht ganz optimalen Ergebnissen führen kann.

Zur Erfassung der Bilder ist es ideal, diese mit der Kamera zu erstellen, mit der wir später auch die Bilderkennung durchführen. So lernt das neuronale Netzwerk auch gleich die Eigenheiten der Kamera mit. Um die Daten mit dem Nvidia Jetson zu erfassen, können wir einen Client für Edge Impulse installieren. Dieser kann dann Bilder (bei Bedarf auch Audio) zu Edge Impulse senden.

Zur Installation des Clients kann man der Anleitung [5] folgen, die Edge Impulse bereitstellt. Für den Nvidia Jetson ist es nötig, sich per SSH mit dem Board zu verbinden und ein paar Befehle auf der Konsole einzugeben. Alternativ kann man auch Maus, Tastatur und Monitor anschließen und die Befehle direkt in die Kommandozeile des Linux auf dem Jetson Nano eingeben.

Wenn eine SSH Verbindung aufgebaut ist oder eine Konsole geöffnet, müssen einfach die Befehle aus der Anleitung eingegeben werden. Für die Installation wird eine Internetverbindung benötigt, sollte der Jetson Nano noch nicht mit dem Internet verbunden sein, muss dies zuerst erledigt werden.

Danach kann in der Konsole

wget -q -0 - https://cdn.edgeimpulse. com/build-system/jetson.sh | bash eingegeben werden, um die Installation von *edge-impule-linux* zu beginnen. Die Installation wird etwas Zeit in Anspruch nehmen. Nachdem die Installation durchgelaufen ist, kann per Konsole <u>edge-impulse-linux</u> clean ausgeführt werden. Das Tool benötigt unsere Zugangsdaten und das Audio- und Video-Device, welches wir für die Datenerfassung verwenden möchten. Ist alles eingerichtet, so erscheint auf der Edge-Impulse-Seite unter *Devices* (**Bild 6**) der neu eingerichtete Jetson Nano. Mit diesem lassen sich nun Bilder von der Kamera aufnehmen.

Auf der Edge-Impulse-Webseite können wir nun unter dem Menüpunkt *Data aquisition* (**Bild 7**) Bilder durch die Kamera erfassen. Da auch eine Audioquelle konfiguriert wurde, muss der Sensor zuerst auf *Camera* (640x480) umgestellt werden. Nun erscheint



I know what I'm doing, hide this wizard!

### Bild 5. Auswahl der Klassifikation.



Bild 6. JetBot als Gerät für die Datenerfassung.

EDGE INFOLSE	tin Did you know? You can	a capture data from any	y device or development boa	rd, or upload y	our existing dataset	s - Show options		
Dashboard Devices	DATA COLLECTED 320 items	•	LABILS 4		0	Record new data		
Data acquisition Impulse design	Collected data			тв	4 0	Device @		
Create impulse	SAMPLE NAME	LABEL	ADDED	LENGTH		Label	Sample length (ms	
<ul> <li>Image</li> </ul>	uno_card_1.jpg.29b69aj0	uno_card_1	Jun 30 2021, 15:27:29		1	Label name	5000	0
Transfer learning	uno_card_1.jpg.29b697ol	uno_card_1	Jun 30 2021, 15:27:26		1			
Retrain model	uno_card_1.jpg.29b694oa	uno_card_1	Jun 30 2021, 15:27:23		1	Sensor	Frequency	
Live classification	uno_card_1.jpg.29b691na	uno_card_1	Jun 30 2021, 15:27:20		1	mulginone		
Model testing	uno_card_1.jpg.29b68trp	uno_card_1	Jun 30 2021, 15:27:16		1		Star	t sampling
Versioning	uno_card_1.jpg.29b68qv2	uno_card_1	Jun 30 2021, 15:27:13		T.			
Deployment	uno_card_1.jpg.29b68o41	uno_card_1	Jun 30 2021, 15:27:10		1	EAW DATA Click on a sample to load		
TING STARTED	uno_card_1.jpg.29b68mh0	uno_card_1	Jun 30 2021, 15:27:08		ι.			
Documentation	uno_card_1.jpg.29b68ki9	uno_card_1	Jun 30 2021, 15:27:06		1			
Forums	uno_card_1.jpg.29b5ls4a	uno_card_1	Jun 30 2021, 15:16:51		1			
	uno_card_1.jpg.29b5lcta	uno_card_1	Jun 30 2021, 15:16:36		1			
	uno_card_1.jpg.29b5l9t3	uno_card_1	Jun 30 2021, 15:16:33		1			

### Bild 7. Datenerfassung mit Edge Impulse.

EDGE IMPULSE	DATA ACQUISITION (UNOCAROS VI						Mathlas ClauBen
	Training data Test data						
<ul><li>Dashboard</li><li>Devices</li></ul>	i Did you know? You can	capture data from any	device or development boa	rd, or upload y	our existing dataset	s - Show options	×
<ul> <li>Data acquisition</li> <li>Impulse design</li> </ul>	320 items	•	LABELS 4		0	Record new data	
Create impulse     image	Collected data			ΥE	a ± 0	Device	~
<ul> <li>Transfer learning</li> </ul>	SAMPLE NAME	LABEL	A0010	LENGTH		Label	Camera feed
× Retrain model	uno_card_1.jpg.29b69aj0	uno_card_1	Jun 30 2021, 15:27:29		I	Label name	
Live classification	uno_card_1.jpg.29b697ol	uno_card_1	Jun 30 2021, 15:27:26		1		Constant -
Model testing	uno_card_1.jpg.29b694oa	uno_card_1	Jun 30 2021, 15:27:23		1	Sensor Camera (640x480)	
Versioning	uno_card_1.jpg.29b691na	uno_card_1	jun 30 2021, 15:27:20		1		
Deployment	uno_card_1.jpg.29b68trp	uno_card_1	Jun 30 2021, 15:27:16		1		Start sampling
GETTING STARTED	uno_card_1.jpg.29b68qv2	uno_card_1	Jun 30 2021, 15:27:13		1		
1 Documentation	uno_card_1.jpg.29b68o41	uno_card_1	jun 30 2021, 15(27)10		1	RAW DATA Click on a sample to load	
Security Forums	uno_card_1.jpg.29b68mh0	uno_card_1	Jun 30 2021, 15:27:08		1		
	uno_card_1.jpg.29b68kl9	uno_card_1	Jun 30 2021, 15:27:06		1		
	uno_card_1.jpg.29b5is4a	une_card_1	jun 30 2021, 15:16:51		1		

Bild 8. Einstellungen für die Kamera.

ein Camera feed, das uns die Live-Bilder der Kamera anzeigt (Bild 8). Unter Label können wir nun eingeben, um welche Art Bild ("Klasse") es sich handelt, also keine Karte, eine Eins, Zwei oder eine Rückseite

Nun kommt der monotone Teil, die Datenerfassung der einzelnen Karten in unterschiedlichen Positionen. Zu Beginn sollten etwa 80 bis 100 Bilder gemacht werden, und das von jeder Karte, die wir später erkennen wollen. Dabei sollte auch gleich das Label angelegt werden, das die aufgenommenen Bilder beschreibt.

## **Erstes Training**

Nachdem alle Bilder erstellt worden sind, ist es Zeit für das erste Training. Unter Impulse design (Bild 9) können wir nun den Impuls festlegen, also die Arbeitsweise des neuronalen Netzwerkes. Für die Bilder verwenden wir 160 x 160 Pixel Größe und Fit shortest axis aus dem Drop-Down-Feld Resize mode. Da keine Farben ausgewertet werden sollen, reicht es als Farbtiefe Grayscale auszuwählen (Bild 10). Nach dem Klick auf Save parameters können auf der folgenden Seite die zu erkennenden Klassen angelegt werden.

Die Punktwolke unter Feature explorer zeigt die Verteilung unserer Bilder anhand eines ersten groben Trainings-Durchlaufs. So lassen sich eventuell falsch beschriftete Daten ausfindig machen. Je dichter die einzelnen Punkte zusammen liegen, umso schwieriger wird es später für das neuronale Netz, die Karten auseinanderzuhalten (Bild 11).

Unter Transfer learning (Bild 12) lassen sich nun Parameter für das Training einstellen. Eingangs wurden Limitierungen der freien Version erwähnt und diese wollen wir hier aufgreifen. Es stehen für das Training 20 Minuten Rechenzeit auf den Servern von Edge Impulse zur Verfügung. Für unsere vier Klassen ist das ausreichend, um brauchbare Ergebnisse zu erzielen. Sollte das Training mehr als 20 Minuten dauern, so wird der Prozess einfach abgebrochen.

Etwas, das die Qualität des Trainings und auch die Zeit beeinflusst, ist die Anzahl der Trainingszyklen. Dieser Wert ist bei den Daten, die angelernt wurden, auf 20 eingestellt. Eine Erhöhung der Zyklen von 20 auf 50 verbessert in dem Fall das Ergebnis. Da der hier verwendete Datensatz sehr überschaubar ist, führt die Verwendung von Data augmentation zu



einer Verbesserung der Erkennung. Bei der Data augmentation werden die Bilder zusätzlich verformt, gestreckt und gedreht, um den Datensatz künstlich zu vergrößern. Dies kann das Ergebnis der Erkennung verbessern. Nachdem das Training abgeschlossen ist, wird eine Statistik erstellt (**Bild 13**), die zeigt, wie gut unser neuronales Netzwerk nun arbeiten würde. Mit dem nun erstellten Netzwerk kann ein Test auf dem Jetson durchgeführt werden.

## **Erster Test auf dem Jetson**

Nachdem das neuronale Netzwerk trainiert ist, muss wieder eine Konsole auf dem Jetson Nano geöffnet werden. Am einfachsten ist es, sich per SSH auf dem Nano einzuloggen. Mit edge-impulse-linux-runner --clean wird beim ersten Start ein Wizard gestartet, der nach den Zugangsdaten fragt. Wenn mehr als ein Projekt angelegt wurde, so muss jenes ausgewählt werden, das gerade getestet werden soll.

Ist alles eingerichtet, so gibt die Konsole bei laufender Kamera eine Reihe von Klassifikationen aus (**Bild 14**). Die Zahlen hinter den einzelnen Klassen geben die Wahrscheinlichkeit wieder, mit der die Klasse in dem gerade aufgezeichneten Bild vorkommt, wobei 1.0 das Maximum (also 100% sicher) und 0.00 das Minimum ist.

Legen wir nun eine Einser-Karte vor die Kamera, dann können wir sehen, wie sich die Wahrscheinlichkeiten verschieben und mit 0.9420 (also einer sehr großen Sicherheit) eine Eins erkannt wurde.

Doch sich nun ganz auf die Konsole zu verlassen ist unpraktisch. So kann es sein, dass die Karte eventuell nicht richtig erkannt wird und vermutet werden muss, was der Jetbot sieht. Das Tool edge-impulse-linux-runner sorgt deshalb nicht nur für die Ausführung des neuronalen Netzwerks, es stellt auch einen Webserver bereit, der auf Port 4912 erreicht werden kann, wie Bild 15 zeigt. Auf der Webseite kann live verfolgt werden, welche Karte gerade erkannt wird, oder wie Bild 15 zeigt, falsch erkannt wird. Die Gelegenheit lässt sich nutzen, um neue Bilder zu generieren, womit das neuronale Netz erneut trainiert werden kann . Dazu muss impulse-linux-runner mit CTRL + C gestoppt werden. Mit edge-impulse-linux erscheint der Jetbot wieder als Datenquelle im Webportal von Edge Impulse und es lassen sich neue Bilder aufnehmen.

## **Training Runde Zwei**

Mit dem Hinzufügen neuer valider Daten kann man ein neues Training beginnen. Je mehr Daten hinzugefügt und passend einsortiert werden, bei denen das aktuelle neuronale Netz noch Probleme hat, umso besser kann es werden. Mit edge-impulse-linux wird der JetBot wieder zum Datensammler; wir können das aktuelle Bild der Karte aufnehmen und passend etikettieren. Wenn das Erfassen von neuen Daten abgeschlossen ist, so kann man unter Retrain Model (Bild 16) das Training mit dem vorherigen Parameter und neuen Daten wiederholen. Nachdem dies abgeschlossen ist, können wir die neue Version des Netzes auf dem Jetson testen. Mit edge-impulse-linux-runner wird das neue neuronale Netz heruntergeladen und auf dem JetBot ausgeführt. Per Webbrowser können wir ansehen, wie die Karte live klassifiziert wird (Bild 17).

## Vom Training zur Anwendung

Nachdem die Daten erfasst wurden und das neuronale Netz trainiert ist, stellt sich die Frage, wie dieses in einer eigenen Anwendung verwendet werden kann. Dazu bietet Edge Impulse ein passendes SDK für unterschiedliche Programmiersprachen an.

Wenn C/C++ verwendet werden soll, kommt beim JetBot leider aktuell ein Problem mit der Kameraansteuerung und der verwendeten OpenMV-Bibliothek zum Tragen, so dass mit dieser Library keine Bilder von der Kamera erfasst werden können (die oben geschilderte Bilderfassung über Edge Impulse ist hiervon nicht betroffen und scheint OpenMV nicht zu verwenden).

Auch wird für die Bildverarbeitung die GPU des Jetson Nano noch nicht genutzt, und damit müssen alle Berechnungen von der CPU erledigt werden. Dies mag für kleine Boards wie einem Raspberry Pi keinen großen Verlust darstellen - beim Jetson Nano liegt damit aber ein großer Teil der Rechenleistung für die KI brach.

Das Training ist aber nicht umsonst gewesen. Das trainierte neuronale Netz kann als plattformunabhängige Bibliothek exportiert werden und somit auch auf anderen Systemen laufen (siehe Download [6]). Dabei muss es nicht ein Raspberry Pi sein, ein ESP32 mit Kamera ist genauso geeignet für das hier trainierte Netz. Leider stehen die reinen Daten, die mit Edge





	NOTHING	UNO_CARD_1	UNO_CARD_2	UNO_CARD_BACK
NOTHING	100%	O 96	0%	0%
UNO_CARD_1	0%	100%	0%	0%
UNO_CARD_2	0%	096	100%	0%
UNO_CARD_BACK	0%	0%	0%	100%
F1 SCORE	1.00	1.00	1.00	1.00







Impulse erfasst wurden, auch nur dort zur Verfügung. Die Bilder, die wir mit Labeln versehen haben, lassen sich nicht exportieren, um sie mit anderen Plattformen oder einem lokalen System zu verarbeiten. Dies ist ein Kompromiss, den man eingeht, wenn man Edge Impulse nutzen möchte.

## Es ist ein Pferd, denn es hat ein Wasserzeichen

Uns Menschen wurde in unserem Leben beigebracht, wie ein Pferd aussieht und woran man es erkennen kann. Dazu wurde uns eine Anzahl an Bildern von Pferden präsentiert und wir haben gelernt, wenn ein Objekt ein Paar Hufe hat, eine bestimmte Größe und Form, dann wird es wohl ein Pferd sein. Und im Laufe unseres Lebens wurde unsere Definition von Pferd immer weiter optimiert: Z.B. ist ein Esel kein Pferd, auch wenn dieser ähnlich wie ein Pferd aussieht. In der klassischen Bildverarbeitung würde man einen Algorithmus lehren, nach welchen Merkmalen er in Bildern zur Erkennung von Pferden suchen muss, zum Beispiel nach Hufen, Augen, Mähne, Farbe, Proportionen. Dies hat aber bei der Generalisierung der Erkennung von Pferden seine Grenzen. Hier kommen die neuronalen Netzwerke zum Zug, man gibt ihnen einfach eine Riesenmenge an Bildern (etliche tausende) und überlässt die Art, wie ein Pferd identifiziert wird, den Algorithmen, die das neuronale Netz prägen. Klingt einfach und was kann schon dabei schiefgehen?

Wie Stuart Cording schon im Webinar "Get to know the NVIDIA Jetson Nano Developer Kit" [7] erzählte, wurde bei einem seiner KI-Projekte zur Erkennung von Pferden auf eine Sammlung von Fotos aus Suchmaschinen zurückgegriffen. Das erzeugt schnell eine passende Menge an Daten; leider jedoch hatten alle Bilder ein Wasserzeichen. Dadurch wurde ein Pferd für das neuronale Netzwerk zu einem Pferd, wenn ein Wasserzeichen in dem zu verarbeitenden Bild zu finden war. Damit schien das Wasserzeichen das prägnanteste Merkmal zu sein, das ein Pferd in diesem Satz Daten ausmachte.

Etwas Ähnliches ist auch bei den ersten Versuchen, Bilder der einzelnen Uno-Karten aufzunehmen, geschehen. Auf einigen Bildern war noch eine Hand zu erkennen; daher wurde die Hand zu einem der prägendsten Mathias Claußen / UnoCards v1

 Image: Claußen

Merkmale, um zwischen der Karte Eins und Zwei zu unterscheiden. Der Datensatz wurde verworfen und passendere Bilder aufgezeichnet. Es gilt also bei der Quelle der Daten aufzupassen, dass nicht ein Wasserzeichen oder ein anderer Gegenstand die Erkennung ungewollt beeinflusst.

## Zusammenfassung und Ausblick

> C @

Q 🔒 10.0.2.130:4912

Edge Impulse und der Jetson Nano machen es leicht, ein eigenes neuronales Netz zu trainieren. Die Einbindung und praktische Anwendung wäre dann der nächste Schritt. Der Jetson Nano ist in der gewählten Konstellation jedoch nicht die am besten unterstützten Hardware, da ein Teil seiner Ressourcen brach liegt und auch die Kamera durch Softwarehürden nicht direkt mit dem SDK von Edge Impulse in eigenen Anwendungen arbeitet. Wer einen einfachen Einstieg in KI und die Bilderkennung sucht, sollte Edge Impulse dennoch probieren. Es muss aber kein Jetson Nano sein, ein Raspberry Pi 4 würde genauso gut seinen Dienst verrichten.

Was kommt also als Nächstes? Der Jetson Nano lässt sich nutzen, um all die Arbeit, die uns Edge Impulse abgenommen hat, auch direkt auf der Hardware auszuführen. Das sollte dann der nächste Schritt werden: das Erstellen eines neuronalen Netzes, das auch Uno-Karten erkennt, direkt auf dem Jetson Nano.

210318-B-02

## Ein Beitrag von

Entwicklung und Text: Mathias Claußen Redaktion: Jens Nickel Layout: Harmen Heida

## 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 oder kontaktieren Sie Elektor unter editor@elektor.com.

## WEBLINKS

[1] Mathias Claußen: "Bildverarbeitung mit dem Nvidia Jetson Nano (Teil 1)", Elektormagazine.de:

www.elektormagazine.de/news/bildverarbeitung-mit-dem-nvidia-jetson-nano-teil-1 [2] Mathias Claußen: "Kaffee zubereiten mit MAX78000 und einer Prise KI", Elektormagazine.de:

www.elektormagazine.de/news/kaffee-zubereiten-mit-max78000-und-einer-prise-ki [3] Edge Impulse: https://www.edgeimpulse.com

[4] Andrea Garrapa: "Edge Impulse: KI für Mikrocontroller", Elektor Industry 2/2021:

www.elektormagazine.de/magazine/elektor-239/59650

[5] Edge Impulse - Nvidia Jetson Nano:

https://docs.edgeimpulse.com/docs/nvidia-jetson-nano

[6] Download des trainierten Netzes: www.elektormagazine.de/210318-B-02

[7] Meet the Engineers (Part 2): www.elektormagazine.com/news/meet-theengineers-2-get-nvidia-jetson-nano-developer-kit

## JUMPA STARTER NEWS o-Kommende Kampagnen

## Von Clemens Valens (Elektor Labor)

Elektor Jumpstarter hilft Innovatoren, ihre kreativen Elektronikprojekte in Produkte zu verwandeln. Mit Elektor Jumpstarter und der Unterstützung unserer Community können Sie Ihren Traum, ein Elektronikprodukt auf den Markt zu bringen, verwirklichen!

## Der (Haus-) Automatisierungscomputer Bumblebee-Automator

Dieses Projekt begann als WLAN-Thermostat, das die Temperatur in einem Weinkeller im Auge behalten sollte. Es wurde in Elektor September 2021 unter dem Titel "ESP32-verbundenes Thermostat" [1][2] veröffentlicht. Während der Vorbereitung des Projekts für die Veröffentlichung wurde schnell klar, dass es viel mehr als nur ein Thermostat ist. Der Autor nutzte es bereits als Luftfeuchtigkeitsregler und ein weiteres Gerät hält das Wasser in seinem Swimmingpool sauber. Bei Elektor haben wir es für präzise Lichtstärkemessungen verwendet und ein serielles Lidar für Entfernungsmessungen und die Erkennung von Einbrechern angeschlossen. Da es sich eher um eine Art (Haus-) Automatisierungscomputer als um ein Thermostat handelt, wurde er in "Automator" umbenannt. Der Automator basiert auf einem frei programmierbaren ESP32-Modul, hat also WLAN- und Bluetooth-Fähigkeiten. Es verfügt über ein Relais, einige konfigurierbare Erweiterungsanschlüsse, und natürlich gibt es auch ein paar LEDs sowie einen Summer. Ein kleines OLED-Display kann einfach angebracht werden, genauso wie die vielen kleinen handelsüblichen Sensor- und Aktormodule mit seriellem, SPI- oder I<sup>2</sup>C-Bus, die alle möglichen Anwendungen erlauben.

Das Gerät besitzt ein integriertes Netzteil, das mit einer Netzwechselspannung von 100...240 V bei 50...60 Hz betrieben werden kann. Da es in ein wasserdichtes Gehäuse passt, kann es auch gut im Freien und in anderen feuchten Umgebungen installiert werden.

Softwareseitig ist der Automator mit ESPHome und somit mit Home Assistant, unserer bevor-



Bild 1. Prototyp des Bumblebee-Automators.

zugten Hausautomations-Controllersoftware kompatibel, aber da es sich im Grunde um einen ESP32 handelt, lässt sich das Gerät in eine Vielzahl von Hausautomationssystemen integrieren. Auch ein Standalone-Betrieb ist natürlich möglich.

Auf der Automator-Grundplatine werden nur bedrahtete Bauteile eingesetzt, so dass sie bei Bedarf leicht modifiziert werden kann.

## Details zur Jumpstarter-Kampagne

Einzelpreis: 59,95 € Ziel: 100 Exemplare Enddatum: 31. Dezember 2021 URL: www.elektormagazine.de/labs/ automator



Bild 2. Prototyp des CO2-Wächters.

## CO<sub>2</sub>-Wächter

Der CO<sub>2</sub>-Wächter [3] überwacht und bewertet der Qualität der Raumluft (IAQ) anhand der Konzentration von Kohlendioxid (CO<sub>2</sub>). Eine Konzentration von weniger als 1.000 ppm (0,1 %) weist unter normalen Bedingungen auf einen hygienisch korrekten Luftaustausch hin und kann dazu beitragen, das Risiko einer COVID-19-Infektion zu minimieren.

Der CO<sub>2</sub>-Wächter hat einen Messbereich von 400...2.000 ppm und zeigt die Ergebnisse in Form einer Ampel an. Grün steht für ausgezeichnete Luftqualität, gelb bedeutet auffällig und rot inakzeptabel. Zwischenwerte werden durch gleichzeitiges Aufleuchten von zwei LEDs angezeigt, wobei grün/gelb für noch akzeptabel und gelb/rot für schon schlecht steht. Ein Summer gibt einen akustischen Alarm, wenn die Konzentration zu hoch wird und Maßnahmen (Öffnen eines Fensters und/oder einer Tür) ergriffen werden sollten.

Der CO<sub>2</sub>-Wächter verwendet einen MH-Z19C-Sensor des Sensorspezialisten Winsen. Dieser Sensor arbeitet mit nichtdispersiver Infrarotdetektion (NDIR), einer häufig verwendeten Technik, die die Menge an Infrarotlicht misst, die von den CO2-Molekülen in der Luft (oder in einem anderen Gas) absorbiert wird. Der Sensor wird von einem ESP8266-basierten Mikrocontroller-Modul mit WLAN ausgelesen. Dieses Modul steuert auch die LEDs und den Summer und liest den Drucktaster aus.

Das Gerät kann sich mit dem Internet verbinden, genauer gesagt mit der ThingSpeak-Plattform, auf der Messwerte aufgezeichnet und grafisch dargestellt werden können. Dies ermöglicht die Überwachung der CO<sub>2</sub>-Konzentration im Zeitverlauf und den Vergleich mit Daten anderer Geräte.

Die Software ist als Arduino-Sketch geschrieben und kann daher leicht modifiziert werden. Der CO<sub>2</sub>-Wächter ist auch mit ESPHome, Home Assistant und anderen Hausautomatisierungsplattformen kompatibel. Der CO<sub>2</sub>-Wächter stellt damit eine preiswerte Möglichkeit dar, die Luftqualität in Klassenräumen zu überwachen und gleichzeitig Schülern den Umgang mit cloud-basierten Plattformen nahebringen.

## **Details zur Jumpstarter-Kampagne**

Einzelpreis: 69,95 € Ziel: 100 Exemplare Enddatum: 31. Dezember 2021 URL: www.elektormagazine.de/labs/co2-guard

210435-03

### **WEBLINKS**

- [1] ESP32 Thermostat bei Elektor Labs: www.elektormagazine.de/labs/esp32-thermostat
- [2] "ESP32-verbundenes Thermostat", Elektor 9-10/2021: https://www.elektormagazine.de/magazine/elektor-182/59862
- [3] CO2-Wächter bei Elektor Labs: www.elektormagazine.de/labs/co2-guard
- [4] Weitere Informationen: www.elektormagazine.de/elektor-jumpstarter



## **Open-Source**-GPS-Tracking-Plattform

Fahrzeug-Tracking ohne fremde Cloud mit Traccar

Von Mathias Claußen (Elektor)

Für das GPS-Tracking von Vehikeln und anderen beweglichen Gütern gibt es eine Menge Hardwaremodule, doch meist muss man die Daten zu einem externen Anbieter in die Cloud schicken, um sie dort zu speichern und zu visualisieren. Die Open-Source-Software Traccar ist eine Lösung, mit der man die Datenhoheit zurückgewinnen kann - sie läuft auf einem Raspberry Pi und arbeitet mit dem Automatisierungsframework Node-RED zusammen.

Jene von uns, die Fahrzeuge tracken müssen oder wollen, zum Beispiel den abgestellten Wohnwagen, können aus einer großen Auswahl an fertigen Trackinggeräten wählen. Die meisten Tracker nutzen das Mobilfunknetz in der 2G- oder 3G-Version. Ein Review eines solchen Trackers [1] haben wir schon durchgeführt. Für diesen Artikel wurden zwei Tracker geordert, einer davon für unter 20 Euro von einem größeren Onlineversandhändler (**Bild 1**). Alternativ lassen sich auch Trackingdaten per LoRaWAN transportieren, wie schon der Elektor LoRa-Tracker [2] gezeigt hat.

Die meisten kommerziellen Tracker bieten dem Benutzer eine Cloud-basierte Lösung, um die Positionsdaten abzurufen. Diese ist für viele Anwender eine komfortable Möglichkeit, sich Daten der Bild 1. Billiger GSM-GPS-Tracker.

Fahrzeuge anzeigen zu lassen und den Bewegungsverlauf und Ereignisse zu verwalten. Bei den Diensten muss aber bedacht werden, wo die Daten liegen. Auch wenn es heißt, dass diese DSGVO-konform verarbeitet werden, muss das noch lange nicht stimmen. Der Benutzer muss darauf vertrauen, dass der Anbieter der gewünschten Rechtsstaatlichkeit unterliegt und kein rechtlich bedenkliches Begehren Dritter mit den Daten befriedigt wird.

## **Installation von Traccar**

Traccar [3] ist eine Software, die auf x86-Rechnern mit Windows und Linux lauffähig ist, sowie Linux-Systemen, die auf ARM-Plattformen laufen, wie dem Raspberry Pi. Es handelt es sich um eine Open-Source-Lösung zum Tracken von Fortbewegungsmitteln. **Bild 2** zeigt die Kartenansicht von Traccar und die konfigurierten Fahrzeuge. Die Software lässt sich aber auch wunderbar zum Tracken von anderen beweglichen Gütern einsetzen.

Die Installation auf einem Raspberry Pi ist denkbar einfach. Es wird ein Raspberry Pi mit Raspberry Pi OS benötigt. Ein Raspberry Pi Model 3B(+) ist für die ersten Versuche ausreichend, ein Raspberry Pi 4 ist empfehlenswert für die Anwendung. Wer keinen Raspberry Pi nutzen möchte, kann Traccar auch auf einem anderen System betreiben.

Von der Traccar-Homepage muss für den Raspberry Pi die ARM-Variante heruntergeladen werden. Auf dem Raspberry Pi kann man dazu einen Webbrowser verwenden. Sollte keine grafische Oberfläche aktiv sein, kann die Datei auch per wget heruntergeladen werden. Der Link zur aktuellen Version ist auf der Traccar-Homepage zu finden. Die heruntergeladene Datei gilt es nun zu entpacken. Dazu können entweder die grafischen Tools des Raspberry Pi OS Desktops verwendet werden oder die Kommandozeile. Sollte es sich bei dem Download um eine Zip-Datei handeln, so lässt sich im Terminal der Befehl unzip nutzen.

Die entpackten Dateien sind in Bild 3 zu sehen. Zur Installation ist nun ein Terminal nötig, da der Installer mit Root-Rechten gestartet werden muss. In dem Terminal wird zu dem Ort navigiert, in dem die entpackten Dateien liegen. Mit sudo ./ traccar.run wird der Installer gestartet. Nachdem die Installation abgeschlossen ist, kann Traccar noch als Service eingerichtet werden, der beim Booten automatisch startet. Dazu ist in dem noch offenen Terminalsudo systemctl enable traccar. service einzugeben. Traccar kann nun mit service traccar start gestartet werden. Nach dem Start steht unter http://localhost:8082/das Webinterface bereit.

Wenn die Geschwindigkeit des Browsers auf dem Raspberry Pi zu träge ist, lässt sich auch von einem anderen Rechner über das Netzwerk auf das Webinterface zugreifen. Für den ersten Login wird als Benutzer admin mit dem Kennwort admin verwendet. Es ist dringendst zu empfehlen, das Passwort gegen ein sicheres auszutauschen. Bild 4 zeigt den Menüeintrag für die Benutzerverwaltung.

Nachdem Traccar läuft, können nun Tracker hinzugefügt werden. Den GSM-basierenden Trackern und deren Datentransport wollen wir uns im nächsten Absatz widmen.

## **GSM-GPS-Tracker**

Die meisten preiswerten GPS-Tracker arbeiten mit einem 2G-Modem für die Nutzung des Mobilfunknetzes. Solche Tracker lassen



Bild 2. Traccar-Oberfläche.



Bild 3. Entpackte Dateien für die Traccar-Installation.



Bild 4. Traccar-Menü für die Benutzerverwaltung.





Bild 5. GSM-GPS-Tracker mit Blick auf die Elektronik.

Bild 6. Datentransport per Mobilfunknetzwerk.



Bild 7. GSM-GPS-Tracker Typ TK103: Antennenanschlüsse.



Bild 8. GSM-GPS-Tracker Typ TK103: Anschlüsse zum Fahrzeug.

sich schon für weniger als 20 Euro (**Bild 5**) erwerben und durch ihre kompakten Abmessungen unauffällig verbauen. Höherpreisige Tracker integrieren teilweise noch ein 3G-Modem, idealerweise aber eines, das schon 4G oder 5G beherrscht.

Ein Tracker, der nur 2G oder 2G und 3G beherrscht, ist auf längere Sicht nicht zukunftsfähig. In der Schweiz ist das 2G-Netz Ende 2020 außer Dienst gestellt worden und die frei gewordenen Frequenzen werden für moderne Übertragungsverfahren genutzt. Für den Rest Europas scheint es noch etwas Schonzeit zugeben, gerüchteweise bis 2025.

So lange kann noch mit den aktuell preiswertesten Trackern gearbeitet werden. Die Funktion dieser Tracker ist simpel. In einem definierten Intervall (oder bei Ereignissen wie Zündung An/Aus) können diese Tracker eine Statusnachricht senden. Dies ist entweder per SMS möglich oder durch ein Datenpaket. **Bild 6** soll den Datentransport verdeutlichen.

Für diesen Artikel betrachten wir den Fall, dass die Daten per Datenpaket gesendet werden. Dazu verwenden die meisten Tracker ein UDP-Paket. UDP-Pakete verlangen im Gegensatz zu TCP-Paketen keine Quittung des Empfängers und reduzieren damit die Datenmenge und die Energie, die zum Übertragen der Statusdaten benötigt werden.

Die Konfiguration der Tracker ist herstellerabhängig und ist aus heutiger Sicht nicht mehr als benutzerfreundlich zu bezeichnen. Es werden bei den meisten preiswerten Geräten SMS-Nachrichten für die Konfiguration verwendet, ähnlich wie beim Netzausfall-Detektor aus Elektor 9/2018 [4]. Je nach Gerät kann das Traccar-Forum [5] Informationen und Hinweise zum Betrieb und der Konfiguration der Geräte liefern.

Damit der Raspberry Pi die Daten der GPS-Tracker empfangen kann, müssen Ports in der Firewall für eine Weiterleitung geöffnet werden. Auch hier hilft ein Blick auf die Traccar-Seite [6], um herauszubekommen, welcher Port benötigt wird. Das Weiterleiten des Ports erfordert je nach Routerhersteller unterschiedliche Schritte. Zusätzlich ist es nötig, dass die externe IP des Routers durch einen DNS-Dienst aufgelöst werden kann; wer keine feste IP-Adresse besitzt, kann auf dynv6, No-IP oder ähnliche dynamische DNS-Dienste zurückgreifen.

## **GSM-Tracker-Testsetup**

Für den Artikel ist ein Test mit einem LK720 GPS-Tracker (Bild 1 und Bild 5) durchgeführt worden. Dieser muss mit 6 V bis 38 V DC versorgt werden und benötigt bei 12 V etwa 150 mA. Die Konfiguration wurde mit Hilfe des Threads [7] aus dem Traccar-Forum versucht.

Leider erweis sich der Tracker als sehr schweigsam, was SMS-Kommandos anging. Eine Konfiguration und Inbetriebnahme waren leider nicht möglich. Ob es sich um ein Montagsgerät oder eine andere Inkompatibilität zwischen GSM-Netz oder SIM-Karte gehandelt hat, lässt sich nicht abschließend beantworten. Wer einen GPS-Tracker haben möchte, bei dem die Funktion nicht Glückssache ist, sollte sich eventuell nach einem anderen Modell umsehen. Als Alternative wurde ein GSM-GPS-Tracker des Typs TK103 bestellt (Bild 7 und Bild 8), der in seinen Abmessungen nicht so kompakt ist wie der LK720 Tracker, dafür aber keine Probleme bei der Kommunikation mit SMS aufweisen sollte. Die Kosten für solche Geräte liegen um die 50 Euro (Versand innerhalb der EU). Der GPS-Tracker wird durch Traccar direkt unterstützt und das Protokoll entsprechend verarbeitet. Es muss der Port 5001 für UDP-Daten im Router geöffnet werden, und zwar zur Weiterleitung an den Rechner, auf dem Traccar läuft.

Die Konfiguration des Trackers erfolgt mit Kommandos per SMS. Die Befehle, nach denen im Handbuch gesucht werden sollte, sind UP, DNS, APN und GPRS. Bei GPRS gibt es zwei Befehlsvarianten, eine zum Einschalten des GPRS-Modus und eine zum Konfigurieren des Protokolls (UDP oder TCP). Setzen Sie den Tracker hier am besten in den UDP-Modus, das hat bei unserem Test dazu geführt, dass Daten übertragen wurden. Für das Tracking wurde ein festes Intervall eingestellt, bei dem alle elf Sekunden die Position übermittelt wird (unabhängig davon, ob sich das Fahrzeug bewegt oder nicht). Ist alles passend eingerichtet, sollten Daten am Traccar-Server ankommen, aber noch keine Anzeige oder Verarbeitung der Daten erfolgen. Es muss noch der Tracker angelegt werden. Dazu wird mit einem Klick auf das Add Symbol (Bild 9) ein neuer Dialog geöffnet, in den der Name des Trackers und seine Kennung eingetragen werden kann (Bild 10). Der Name kann frei vergeben werden; für die Kennung muss beim TK103 die IMEI des Gerätes eingegeben werden. Diese kann per SMS abgefragt werden, die Details dazu sollten im Handbuch des Trackers zu finden sein. Sind diese Daten eingetragen, dann sollte der Tracker in der Liste der Geräte erscheinen und der Eintrag grün hinterlegt sein (Bild 11). Auf der Karte wird nun auch die aktuelle Position des Trackers angezeigt.

Traccar zeichnet nun die Bewegungen und Ereignisse des Trackers auf. Bild 12 zeigt den Tracker in einer portablen Kiste, fertig für eine kurze Testrunde. Das Ergebnis lässt sich in **Bild 13** betrachten: Neben der Position werden auch Ereignisse vom Tracker gemeldet und von Traccar verarbeitet. Eine Auswertung ist in Bild 14 zu sehen.

Doch was ist, wenn man mit einem GSM-gestützten Tracker nicht zufrieden ist, oder wenn man mehr Informationen wie zum Beispiel OBD-Fahrzeugdaten auswerten möchte? Dann kann man sich auch seinen eigenen GPS-Tracker bauen und die Daten an Traccar schicken.

## Alternative Trackinggeräte und Node-RED

An Traccar kann man nicht nur GSM-gestützte Tracker, sondern universell Trackinggeräte anbinden. Zum Beispiel Tracker, die nicht per GSM kommunizieren, sondern Sigfox oder LoRaWAN nutzen. Wenn Geräte nicht direkt Daten an den Server senden können (wie bei Sigfox oder LoRaWAN), ist ein Gateway nötig. Dieses übersetzt dann die Daten in etwas, das Traccar leicht versteht und liefert diese Daten an. OsmAnd ist eines der Protokolle, die Traccar versteht; es ist überdies leicht umzusetzen. Das OsmAnd-Protokoll arbeitet mit http-POSTs, um Daten zu übertragen. Das ist etwas, das mit dem



Bild 9. Hinzufügen eines Gerätes in Traccar.

Name	Status	Last Update	
Mobile	Unknown	3 days	
ТК103А	Unknown	15 hours	
	Device		×
	- Required		_
	Name:		
	ldentifier:		
	— 🛛 Extra		-
	Attributes	<b>~</b>	×

Bild 10. Dialog für die Basisdaten.



Bild 11. Tracker ist online.



mächtigen Automatisierungsframework Node-RED recht schnell generiert werden kann.

Um Node-RED auf dem Raspberry Pi zu installieren, kann man der Installationsanleitung [8] für den Raspberry Pi folgen. Nach der Installation kann Node-RED mit node-red-start aus einem Terminal heraus gestartet werden. Wenn nun Traccar und Node-RED als Tandem laufen, lassen sich Daten beliebig modifizieren und weiterverarbeiten. Der Ablauf ist in **Bild 15** noch einmal dargestellt.

Ein Beispiel für ein nicht direkt unterstütztes Gerät ist der LoRa-GPS-Tracker von Elektor [9]. Für den LoRa-Tracker gibt es bereits einen Node-RED-Flow, der die Positionsdaten in einer OpenStreet-Map-Karte sichtbar macht. Diese Variante hat aber für das Tracking von Geräten gewisse Limitierungen und zeigt auch keine Routen an. Doch nun können wir Traccar plus Node-RED verwenden, um Daten von LoRaWAN-Trackern auszuwerten und komfortabel anzuzeigen. Doch nicht nur das, durch Node-RED erhält das System eine universelle Datenanbindung. Es lassen sich auch Positionsdaten von Wetterballons und vieles mehr tracken.

Ein Beispielflow für Node-RED, mit dem man Daten an Traccar senden kann, wird mit diesem Artikel mitgeliefert [10]. Wenn Sie mehr über Node-RED in Verbindung mit einem Raspberry Pi erfahren möchten, schauen Sie doch mal in das Buch *Programming with Node-Red* [11] aus dem Elektor-Store.

## Ausblick

Herr der eigenen Daten zu sein, ist in diesen Zeiten sicherlich nicht verkehrt. Der eine oder andere Betreiber von Webseiten, die personenbezogene Daten speichert, musste in jüngster Zeit feststellen, dass diese entweder abhandenkommen [12] oder dass die Cloud sich einfach in Rauch auflöst [13]. Hinzu kommt mehr Flexibilität, die Daten weiterzuverarbeiten oder das Tracking für andere Einsatzzwecke auszubauen. Wenn man die Anwendung zum Beispiel mit dem Projekt *ResQ Search and Rescue Tools* auf unserer Projektplattform *elektor-labs* [14] verbindet, lassen sich die entsprechenden Daten mit etwas Anpassung importieren



Bild 12. Box mit Tracker und Akku.



Bild 13. Routenaufzeichnung mit Traccar.







und verarbeiten. So könnte eine Trackingkarte von vermissten Personen erstellt werden, wenn man diese per Bluetooth oder WLAN ortet.

Mit einem Raspberry Pi ist der Energieverbrauch für das Tracken überschaubar. Und sollte auch der Raspberry Pi Gefahr laufen, entwendet zu werden, so lassen sich die Daten mit sich selbst verschlüsselnden SD-Karten [15] vor unbefugtem Zugriff schützen. 

200532-02

## **Ein Beitrag von**

Text: Mathias Claußen Redaktion: Jens Nickel Layout: Harmen Heida, Giel Dols

## Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter mathias.claussen@elektor.com

## **PASSENDE PRODUKTE**

- > Raspberry Pi 4 B (4 GB RAM) (SKU 18964) www.elektor.de/18964
- > Offizielles EU-Netzteil für Raspberry Pi 4 (weiß) (SKU 18962) www.elektor.de/18962
- > JOY-iT Acrylgehäuse mit Duallüfter für Raspberry Pi 4 (SKU 19053) www.elektor.de/19053

## WEBLINKS .

- [1] Review: Fortebit Polaris 3G kit+, Elektormagazine.de: www.elektormagazine.de/news/fortebit-polaris-3g-kit
- [2] LoRa GPS Tracker, Elektor Labs: www.elektormagazine.com/labs/lora-gps-tracker
- [3] Traccar Homepage: www.traccar.org/
- [4] Netzausfalldetektor mit SMS-Alarm, Elektor 9/2018: www.elektormagazine.de/magazine/elektor-59/41883
- [5] Traccar Forum: www.traccar.org/forums
- [6] Ports für diverse GPS-Tracker: www.traccar.org/devices/
- [7] Hinweise zur Konfiguration des CJ720: www.traccar.org/forums/topic/lk720-also-known-as-cj720/
- [8] Node-RED Installationsanleitung: https://nodered.org/docs/getting-started/raspberrypi
- [9] LoRa-GPS-Tracker, Elektor 11-12/2020: www.elektormagazine.de/magazine/elektor-158/59064
- [10] Node-RED Flow Download: https://bit.ly/3yHYPKW
- [11] Dogan Ibrahim: Programming with Node-RED, Elektor: www.elektor.de/programming-with-node-red-e-book
- [12] Facebook Informationen zum Datenleck: https://datenschutz-hamburg.de/pages/fb-leak/
- [13] Brand bei OVHcloud : www.reuters.com/article/us-france-ovh-fire-idUSKBN2B20NU
- [14] ResQ Search and Rescue Tools, Elektor Labs: www.elektormagazine.de/labs/resq-search-and-rescue-tools
- [15] Secure Boot für Raspberry Pi, Elektormagazine.de:

www.elektormagazine.de/news/translation-thomas-secure-boot-solution-for-raspberry-pi



## Multifunktions-Komponententester LCR-T7 von JOY-iT



## Testen von passiven Bauteilen, diskreten Halbleitern und IR-Fernbedienungen

## Von Luc Lemmens (Elektor)

## Suchen Sie nach einem einfachen, aber praktischen Multifunktionstester? Der LCR-T7 von JOY-iT könnte *die* Lösung für Sie sein!

Multifunktionstester sind hervorragend zum Test passiver und aktiver diskreter Bauelemente geeignet. Manchmal sind Werte oder Typennummern schwer zu lesen, und bei diskreten Halbleitern mit exotischen Typennummern möchten Sie vielleicht erst einmal feststellen, um welche Art von Bauteil es sich handelt. Und selbst wenn Sie das Bauteil visuell identifizieren können, möchten Sie doch gerne wissen, ob es noch funktioniert und ob es - bis zu einem gewissen Grad seinen ursprünglichen Spezifikationen genügt. Da kommt ein Tester wie der LCR-T7 JOY-iT gerade recht: eine schnelle Überprüfung, mit welchem Bauteil wir es zu tun haben, oder zumindest um zu sehen, ob es das Bauteil ist, das wir erwartet haben.

Mit anderen Worten: Mit dieser Art von Geräten ist es einfach, diskrete Bauteile zu testen, festzustellen, um welche Art von Bauteil es sich handelt, und - im Falle von Halbleitern - sogar die richtige Anschlussbelegung herauszufinden. Wie der Name schon sagt, handelt es sich um ein Prüfgerät. Erwarten Sie also nicht, dass Sie damit hochpräzise Messungen durchführen können, aber das sollte man von einem Prüfgerät, das weniger als drei Zehner kostet, auch nicht verlangen. Das Schöne am LCR-T7 ist, dass es sehr einfach zu bedienen ist: Schließen Sie das zu prüfende Bauteil (DUT) an der ZIF-Fassung auf der Vorderseite an, entweder mit oder ohne die mitgelieferten Drähte und Klemmen, und drücken Sie die Start-Taste. Das war's auch schon. Das Gerät erkennt automatisch, welche Art von Bauteil angeschlossen ist, und zeigt dessen Hauptparameter und Anschlussbelegung (falls zutreffend) im LC-Display an.

## **Unterstützte Bauteile**

Die Liste der Bauteile (siehe Textkasten), die mit dem Joy-IT LCR-T7 getestet werden können, ist für ein so kleines und preiswertes Gerät recht beeindruckend. Die Werte, die gemessen und angezeigt werden, hängen natürlich von der Art des zu prüfenden Bauteils ab. **Bild 1** zeigt einige Ergebnisse für verschiedene Arten von Bauteilen, das Handbuch [1] bietet eine vollständige Übersicht über alle Bauteile, die getestet werden können. Obwohl Darlington-Transistoren nicht aufgeführt sind, scheint der LCR-T7 dennoch einige nützliche Informationen auch über diese Bauteile zu liefern. Sie sind an einer hohen U<sub>be</sub> (> 0,7 V) zu erkennen, und die Kennzeichnung der Anschlüsse ist bei den von mir getesteten Darlingtons korrekt. Allerdings sollte man den anderen Messwerten, die für diesen Transistortyp angezeigt werden, nicht trauen.

## **Erste Schritte**

Nach dem Auspacken muss als Erstes der interne Lithium-Akku über das mitgelieferte Micro-USB-Kabel aufgeladen werden. Die zweifarbige LED neben dem USB-Anschluss leuchtet während des Ladevorgangs rot und wird grün, wenn der Akku vollständig geladen ist. Das Prüfgerät kann auch verwendet werden, wenn es über USB mit Strom versorgt wird. Führen Sie eine automatische Kalibrierung des LCR-T7 durch, indem Sie die Klemmen kurzschließen (**Bild 2**) und die Starttaste drücken. Wenn etwa ein Viertel der Kalibrierung abgeschlossen sind, werden Sie auf dem Display aufgefordert, die Kurzschlüsse zu entfernen. Ist dies geschehen, wird die Kalibrierung automatisch fortgesetzt und beendet. In der Bedienungsanleitung wird nicht erwähnt, wann oder wie oft der Tester neu kalibriert werden muss, aber es kann nicht schaden, dies von Zeit zu Zeit zu tun, vor allem, wenn Sie sich über die Testergebnisse eines Bauteils nicht sicher sind.

Für erste Tests werden ein Elektrolytkondensator und eine LED im Paket mitgeliefert. Meiner Meinung nach wäre es besser gewesen, einige dreibeinige diskrete Halbleiter beizulegen, um die automatische Bauteiltyp- und Pin-Erkennung zu demonstrieren.

## **Anschluss von Bauteilen**

Eine ZIF-Buchse für den Anschluss von Prüflingen (Bild 3) ist eine clevere Lösung, denn einfache, normale Buchsen, wie sie bei vielen ähnlichen Prüfgeräten zu finden sind, nutzen sich im Handumdrehen ab. Für den Anschluss von Bauteilen mit dickeren Stiften werden drei Kabel mit Prüfklemmen mitgeliefert, um eine Beschädigung der Kontakte der ZIF-Buchse zu vermeiden. Für die meisten Bauteiltests werden die komplette obere Reihe und die vier Anschlüsse ganz rechts in der unteren Reihe verwendet, wobei 1, 2 und 3 die Position für die Anschlüsse des Prüflings markieren. Die drei verbleibenden Stifte links in der unteren Reihe (gekennzeichnet mit K und zweimal A) sind für die Prüfung von Zenerdioden mit Spannungen bis zu 30 V vorgesehen.

## **IR-Detektor**

Das LCR-T7 verfügt auch über einen integrierten IR-Detektor, der die Geräte- und Datencodes von Fernbedienungen mit NEC-Protokoll dekodieren und anzeigen kann. Bei anderen IR-Protokollen zeigt das Gerät mit einem roten Punkt in der Ecke des Bildschirms, dass ein IR-Signal erkannt wird, so dass Sie zumindest überprüfen können, ob Ihre Fernbedienung prinzipiell funktioniert oder nicht.

## Einfach und doch nützlich

Trotz seiner Einfachheit ist das LCR-T7 ein sehr nützliches Gerät. Für die Messung von beispielsweise einer Batteriespannung oder eines Widerstandes ziehe ich ein normales Multimeter vor, vor allem, weil es ohnehin immer auf meinem Labortisch bereit liegt. Aber bei anderen Bauteilen, vor allem bei Halbleitern, ist es wirklich großartig, schnell den Typ und die Anschlussbelegung zu identifizieren und die wichtigs-



> JOY-iT LCR-T7 Multifunktions-Komponententester (SKU 19709) www.elektor.de/19709

ten Parameter durch einfaches Anschließen und Drücken einer Taste auf das Display zu bekommen. Und was uns allen im Labor gefällt: Der Preis ist auch nicht schlecht! .∎

210365-02

## **Eigenschaften und Spezifikationen**

WICHTIGSTE EIGENSCHAFTEN				
Messbare Bauteile	Widerstand, Kondensator,			
	Induktivität, Thyristor, Triac,			
	(Doppel-) Diode, Z-Diode,			
	Feldeffekttransistor, Bipolartransis-			
	tor, Infrarot-Fernbedienung			
Unterstütztes IR-Protokoll	NEC			
	(von vielen Herstellern verwendet)			
Anzeigeart	1,8"-TFT-LCD (160 x 128 Pixel)			
Besondere Merkmale	Automatische Kalibrierung,			
	Ein-Tasten-Bedienung			
Eingebaute Batterie	Lithium-Ionen-Akku, 3,7 V, 350 mAh			
MESSBEREICHE				
Kapazität	25 pF100 mF			
Widerstand	0,01 Ω50 ΜΩ			
Induktivität	0,01 mH20 H			
Batteriespannung	0,14,5 V			
Z-Diode	U <sub>BR</sub> 0,0130 V			
Diode	U <sub>F</sub> < 4,5 V			

### **WEBLINK**

[1] SIMAC Electronics (JOY-iT), "Multi-Funktions-Tester", 2021: https://bit.ly/3cY0Ppc



Bild 2. Kalibrierung mit kurzgeschlossener Prüfbuchse.



Bild 3. Testen von Bauteilen in der ZIF-Fassung oder mit den Testclips.





Sehen Sie sich dieses Elektor-Projektvideo auf youTube an



## Rauschen zur Musik mit dem PRBSynth1

## Von Raymond Schouten (Niederlande)

In der elektronischen Musik und bei Soundeffekten ist die Verwendung von Rauschen weit verbreitet. Der hier beschriebene Synthesizer lädt dazu ein, rauschbasierte Klänge zu erzeugen und zu erforschen. 2 Die Hardware ist recht einfach, im Grunde nur ein Arduino Pro Mini. erweitert um einen DAC, einen Lautstärkeregler, eine Stereo-Panning-Schaltung sowie einen MIDI-Eingang und -Ausgang. Die meisten Funktionen sind in der Software implementiert, angefangen von der Klangerzeugung über vier verschiedene Filtertypen bis hin zu einem mächtigen LFO, der den Klang auch mit rhythmischen Patterns steuern kann.





Der PRBSynth1 ist ein Musiksynthesizer für die Erzeugung, Formung und Modulation von rauschbasierten Klängen. Es können viele "klassische" Synthesizer-Sounds erzeugt werden, aber das Gerät erlaubt auch die Erkundung anderer, eigener klanglicher Wege. You can go your own way!

Aus diesem Grund soll dieser Artikel nicht nur den Synthesizer beschreiben, sondern Sie auch einladen, mit ihm zu experimentieren, indem Sie ihn entweder anpassen oder einige der hier vorgestellten Ideen verwenden (siehe Textkästen). Sie könnten damit beginnen, eine vereinfachte Version auf einem Breadboard aufzubauen, ohne MIDI und ohne Panorama-Schaltung mit ihren Potentiometern und LEDs. Sie können auch einen Arduino Uno verwenden, auch wenn dieser nur sechs analoge Eingänge hat, was bedeutet, dass Sie auch nur sechs Potentiometer einsetzen können. Eine andere Möglichkeit ist, eine reine MIDI-Version ganz ohne Potentiometer zu bauen und den Teil des Programms zu deaktivieren, der sie ausliest.

## **Die Sound-Engine**

Die Architektur des PRBSynth1 ist, wie Bild 1 zeigt, recht klassisch. Ein Sound wird durch das Einstellen von acht leistungsstarken Funktionen erzeugt. Diese Funktionen werden entweder mit acht Potentiometern oder über MIDI gesteuert. Der Synthesizer antwortet auch auf MIDI-Note-On- und Note-Off-Befehle mit Velocity-Daten. Ein neuntes Poti mit eingebautem Schalter stellt den Ausgangspegel ein und schaltet die interne Batterie ein und aus. Ein Display ist nicht erforderlich, denn die aktuelle Betriebsart des Synthesizers wird durch einige LEDs deutlich angezeigt.

## **Rauschgenerator**

In einem herkömmlichen Synthesizer wird der Klang in Form von Sinus-, Sägezahn- oder Rechteckwellen durch einen Oszillator erzeugt. In diesem Synthesizer geschieht dies durch einen Rauschgenerator, der vier Arten von Rauschen erzeugt: weiß, rosa, rot und Popcorn. Weißes Rauschen besitzt ein horizontales Spektrum, was bedeutet, dass jede Frequenz die gleiche Intensität hat. Das ist ähnlich wie beim weißem Licht, weshalb es auch weißes Rauschen genannt wird. Es klingt wie ein Radio, das zwischen zwei Sendern abgestimmt ist.



Bild 1. Funktionsübersicht des PRBSynth1. Jeder Block verfügt über einen oder mehrere Regler in Form eines Potentiometers oder eines MIDI-CC-Parameters. Die orangefarbenen Punkte stellen LEDs dar.

Durch die Betonung bestimmter Teile des Spektrums können andere Rauschfarben erzielt werden. So steigt beim 1/f- oder Rosa Rauschen das Spektrum mit -10 dB pro Dekade, bei Brownschem oder 1/f2- oder Rosa Rauschen sind es -20 dB pro Dekade. Popcorn-Rauschen entsteht durch Übersteuerung am Ausgang des Rauschgenerators. Das Rauschen wird digital in der Software von einem "abgegriffenen" Schieberegister mit EXOR-Rückkopplung erzeugt, wie dies im Textkasten näher erläutert wird. Wählt man die Rückkopplungsabgriffe an den richtigen Stellen, wird der Ausgangsbitstrom fast zufällig. Ein solcher Bitstrom wird als pseudozufällige Binärfolge (PRBS) bezeichnet. Er ist nur scheinbar zufällig, weil sich der Ausgangsbitstrom nach einer bestimmten Zeit wiederholt. Wenn das Schieberegister lang genug ist, wird auch die Periode des Bitstroms sehr lang, und man kann ihn als Rauschen betrachten. Technisch gesehen ist ein solcher Rauschgenerator also eigentlich ein Oszillator mit einer sehr komplexen Wellenform und einer sehr langen Periode.

Mit dem PRBSynth1 können Sie auch die Länge der Zufallszahlenfolge steuern, um einen allmählichen Übergang von Rauschgeräuschen zu sich wiederholenden Tonhöhenmustern zu erreichen. Dies wird als Pattern-Modus bezeichnet. Die Tonhöhe dieser Muster wird auf Musiknoten abgebildet und kann mit einem MIDI-Keyboard gespielt werden.

### Filter

Das Ausgangssignal des Rauschgenerators wird in ein Filter eingespeist. Dabei handelt es sich um eine Software-Implementierung

eines Multimode-Filters (siehe Textkasten "Software-Filter leicht gemacht") mit den Modi Tiefpass, Bandpass, Hochpass und Bandsperre (Notch). Die Tiefpass- und Hochpassfilter haben jeweils zwei Flankensteilheiten: erste Ordnung mit 6 dB pro Oktave oder zweite Ordnung mit 12 dB pro Oktave. Es kann immer nur ein Filter verwendet werden. Die Eckfrequenz ist mit einem Potentiometer einstellbar, nicht aber die Resonanz (auch bekannt als Güte oder Q-Faktor). Stattdessen haben die meisten Filter mehrere Q-Voreinstellungen: drei für das 12-dB-Tiefpassfilter, fünf für das Bandpassfilter, während das 12-dB-Hochpassfilter und das Notch-Filter jeweils zwei Q-Voreinstellungen besitzen. Das ergibt insgesamt 14 Filtervoreinstellungen. Ein fünfzehntes Filter ist ein Tiefpassfilter mit fester Frequenz, aber modulierter Abtastrate.

Drei LEDs zeigen an, welche Filtervariante aktiv ist. Ihre Helligkeit ist ein Maß für die Grenzfrequenz.

## Verstärker

Auf den Filterausgang folgt ein Verstärker. Dem aufmerksamen Leser dürfte aufgefallen sein, dass die bei (analogen) Synthesizern typischen Hüllkurvengeneratoren (ADSR) fehlen. Der PRBSynth1 hat keine, aber man kann den LFO in einen einfachen Hüllkurvengenerator verwandeln, indem man ihn in den Trigger-Modus versetzt (siehe unten). Wenn der LFO durch eine Taste oder den Trigger-Schalter ausgelöst wird, führt er einen einzigen Zyklus aus. Das LFO-Wellenform-Potentiometer bestimmt die Form der Hüllkurve, während das Rate-Potentiometer die Dauer festlegt. Diese Hüllkurve kann die





Bild 2. Eine detaillierte Übersicht der Reglerfunktionen des Synthesizers. Einige der Potentiometer fungieren teilweise als Multipositionsschalter und teilweise als normale Potentiometer. Die Skalen sind die gleichen, wenn Sie die entsprechenden MIDI-CC-Controller verwenden. Eine größere Zeichnung kann von der Labs-Webseite [1] heruntergeladen werden.

Filter-Eckfrequenz oder die Amplitude des Signals steuern. Darüber hinaus schaltet der Verstärker das Signal durch, wenn eine (MIDI-) Taste gedrückt wird, so dass Sie den Klang durch Loslassen der Taste abschneiden können.

Die Lautstärke des digitalen Signals kann über MIDI gesteuert werden, was zwar eine Anpassung der Gesamtlautstärke ermöglicht, aber nicht empfohlen wird. Um eine optimale Signalqualität zu erzielen, sollte man immer die maximale digitale Signallautstärke anstreben. Zur Einstellung des analogen Ausgangspegels des Synthesizers steht ein Potentiometer hinter dem Digital-Analog-Wandler (DAC) zur Verfügung.

## **Stereo-Panning**

Obwohl das Ausgangssignal des Synthesizers monophon ist, kann es im Stereobild (Panorama) verschoben (gepannt) werden. Die Ausgangsbuchse des Synthesizers ist daher eine Stereo-Ausführung. Mit einem einzigen Potentiometer können Sie das Signal manuell von links nach rechts positionieren oder entweder den LFO oder einen speziellen freilaufenden Modulator mit eigenem Rate-Regler steuern. Zwei LEDs zeigen an, was vor sich geht.

## Niederfrequenz-Oszillator

Ein Low-frequency-Oscillator (LFO) bringt etwas Abwechslung in die statischen Synthesizer-Sounds. Wenn Sie den LFO langsam über die Eckfrequenz des Filters wandern lassen, entstehen beispielsweise Klänge wie Wind, Meer und Regen. Der LFO kann auch die Signalamplitude, die Länge des rauscherzeugenden Schieberegisters und die Stereoposition des Ausgangssignals modulieren. Die LFO-Frequenz ist von 25 Hz bis zu einem Zyklus alle zwei Minuten einstellbar. Neben diesem Rate-Regler verfügt der LFO über 16 verschiedene Wellenformen, die von einigen Sägezahnvarianten und einem Dreieck bis hin zu einigen Impuls- und Zufallswellenformen reichen. LFO-Geschwindigkeit und Wellenform werden durch eine einzige LED angezeigt.

Der LFO verfügt über vier Betriebsmodi, die über ein Potentiometer gesteuert werden. Wenn das Potentiometer ganz nach links gedreht ist, ist der LFO "klassisch" die ganze Zeit aktiv. Dies ist der erste Modus. Die zweite Möglichkeit besteht darin, "Akzente" aus einer Mustertabelle hinzuzufügen. Mit dem LFO-Mode-Potentiometer können Sie zwischen 48 rhythmischen Patterns wählen, die meisten mit einer Länge von acht Schritten und zwölf mit einer Länge von sechs. Der dritte Modus bietet 16 LFO-Depth-Sweep-Patterns, drei davon mit einer Länge von zwölf Takten, die anderen mit einer Länge von 16 Takten. Der LFO-Mode-Regler wählt das Sweep-Muster aus.

Wenn der Mode-Regler schließlich ganz nach rechts gedreht ist, wird der vierte Modus aktiviert. In diesem Modus führt der LFO nach dem Drücken der Trigger-Taste oder einer Taste auf dem MIDI-Keyboard genau einen Zyklus aus. Auf diese Weise können Sie die Hüllkurvenmodulation manuell anwenden.

## **Benutzeroberfläche**

Es gibt also eine Vielzahl von Klangparametern, die vom Benutzer gesteuert werden können, aber nur acht Potentiometer (vom Ausgangslautstärke-Poti einmal abgesehen). Um diese Regler optimal nutzen zu können, sind fast alle in Segmente unterteilt. Nur die Regler für die Filter-Eckfrequenz und die LFO-Rate funktionieren wie normale Potentiometer, die anderen werden eher wie Schalter mit mehreren bis vielen Positionen angewendet. Aus diesem Grund ist eine MIDI-Bedienoberfläche praktisch, da sie oft eine digitale Anzeige für CC-Werte (Continuous Control) hat und man so weiß, wo genau man sich auf der Skala eines Reglers befindet. **Bild 2** zeigt, wie die verschiedenen Funktionen auf die Potentiometer aufgeteilt sind.

Ein Beispiel: Beim Rauschquellen-Potentiometer (MIDI-CC-Code 75) fungiert das Segment von -135° bis -45° (MIDI-CC-Werte 0...39) als Schalter, mit dem Sie zwischen den verschiedenen Rauschfarben wählen können. Dies wird als *Noise Mode* bezeichnet. Von -45° bis +90° (MIDI-CC-Werte 40...111) bestimmt dieses Potentiometer die Länge des Schieberegisters. Der Synthesizer befindet sich dabei im *Pattern Mode*. Das letzte Segment von +90° bis +135° (MIDI-CC-Werte 112...127) regelt die LFO-Modulationstiefe mit der Länge des Schieberegisters.

Sicherlich ist dies nicht die intuitivste Oberfläche, und Sie werden sich anfangs häufig an Bild 2 orientieren müssen, aber einige LEDs helfen, bestimmte Einstellungen zu erkennen. Eine größere Zeichnung kann von [1] heruntergeladen werden. Eine andere Lösung ist die Herstellung einer Frontplatte, auf der diese Informationen um die Regler herum aufgedruckt sind. Drei LEDs zeigen den gewählten Filtertyp an; ihre Helligkeit die Filter-Grenzfrequenz. Die LFO-LED zeigt die LFO-Rate an, während ihre Intensität auf die gewählten Wellenform hinweist. Die Stereo-Position des Signals wird durch zwei besondere LEDs angezeigt. Und schließlich gibt es noch einen Trigger-Schalter, der entweder einen einzelnen LFO-Zyklus startet (im Trigger Mode) oder die LFO-Accents verändert (im Accent Mode), was für rhythmische Klänge interessant ist.

## Ein Blick auf die Schaltung

Der Schaltplan des PRBSynth1 (**Bild 3**) ist recht einfach, da die meisten Funktionen in der Software implementiert sind. Im Grunde handelt es sich um einen Arduino Pro Mini, der um einen DAC, einen Lautstärkeregler und eine Stereo-Panorama-Schaltung sowie einen MIDI-Ein- und -Ausgang erweitert wurde.

## Ein 7,75-Bit-D/A-Wandler

Auf den ersten Blick scheint der DAC, der das digitale Audiosignal in ein analoges umwan-



Bild 3. Ein Arduino Pro Mini, ein Optokoppler und eine Handvoll passiver Bauteile stellen die komplette Hardware dar. Besonders erwähnenswert sind die Stereo-Panning-Schaltung (D1, D2) und der 6-Bit-BiTriDAC mit R10...R15.



Bild 4. Die Sound-Engine wird von Timern gesteuert. Der hochfrequente Timer1 erzeugt Samples, die vom niederfrequenten Timer0 moduliert werden. Timer2 steuert das Stereo-Panning. Die (nicht abgebildete) MIDI-Datenverarbeitung findet statt, wenn keine Interrupts bedient werden.

delt, ein einfaches gewichtetes 6-Bit-Widerstandsnetzwerk zu sein, aber in den Widerstandswerten und der Art, wie sie angesteuert werden, sind einige Tricks versteckt. Dieser 6-Bit-DAC kann nämlich 216 Ausgangswerte anstelle von nur 64 erzeugen. Die unteren drei Bits (LSBs) sind "trinär" und können nicht nur zwei, sondern drei Werte annehmen (Null, offen und Eins). Die oberen drei Bits (MSBs) sind dagegen normale binäre Bits mit zwei Werten. Dieser (von mir so genannte) BiTri-DAC hat also die etwas ungewöhnliche Auflösung von 7,75 Bit.

Der Nachteil ist eine etwas höhere Komplexität der Software, da die 8-Bit- auf 7,75-Bit-Werte abgebildet werden müssen und weil das Setzen eines trinären Bits etwas aufwändiger ist als die Ausgabe eines binären Bits. Aber die Mapping-Funktion sorgt für ein sanftes Clipping und verleiht dem Klang die gewünschte analoge Anmutung.

Um die beste Leistung des DACs zu erzielen, ist es sehr empfehlenswert, die LED des Arduino Pro Mini zu entfernen, die an Pin 13 angeschlossen ist (neben dem Reset-Knopf, das Entfernen des Vorwiderstands könnte einfacher sein), da sie parallel zu R10 des DACs liegt. Dies verbessert nicht nur die Präzision des DAC, sondern hilft auch, etwas Strom zu sparen.

## Spannungsgesteuerter Stereo-Panner

Die beiden Dioden D1 und D2 fungieren als variable Wechselstromwiderstände, die das kleine Audiosignal (weniger als 200 mV) für den linken (D2) und rechten (D1) Kanal abschwächen. Beide Dioden werden durch den Schleifer von P1 (Lautstärke) vorgespannt, der gemittelt auf 2,5 V steht. Die Abschwächung wird von den Gleichströmen durch die Dioden bestimmt, die von der Spannung am Knoten R5/C6/D1/D2 beeinflusst werden. Diese Spannung ist eine gefilterte Version des PWM-Signals an Pin 3 des Arduino Pro Mini. Durch einen ungleichen Steuerstrom zu den Dioden wird der Ton im Stereofeld beweat. Um den Effekt nicht ausarten zu lassen, ist das Panning auf eine Unsymmetrie von etwa 10 dB begrenzt.

Streng genommen handelt es sich bei dieser Schaltung nicht nur um einen reinen Stereopanner, da sie auch eine Amplitudenmodulation (Tremolo) hinzufügt, während sie den Klang bewegt. Für eine Synthesizer-Anwendung kann dies als Bonus betrachtet werden. Dasselbe gilt für die Verzerrung, die dem

## Software-Filter leicht gemacht

Software-Filter können analoge Filter nachahmen. Das Programm folgt einfach dem Signalfluss seines analogen Gegenstücks und benötigt nur vier Codezeilen:

LP = LP + F\*BP; // F is the filter Frequency setting HP = Vin - LP - q\*BP; // note: q = 1/Q, avoids a divisior BP = BP + F\*HP; DAC = LP; // or select DAC=HP, DAC=BP

Was ist mit den beiden Integratoren? Der Code läuft mit einer bestimmten Geschwindigkeit ab. Daher ist ein analoger Integrator gleichbedeutend mit der Addition einer Variablen zu ihrem vorherigen Wert, wie dies in der ersten und der dritten Zeile des Codes geschieht. Ein 8-Bit-Prozessor mit 2 KB Speicher für die Berechnung eines Softwarefilters erfordert einen kompakten und schnellen Code für die Audioverarbeitung. Dies wird durch die Verwendung von Byte- und in einigen Fällen von 16-Bit-Integer-Variablen erreicht. Gebrauchen Sie keine Divisionen in der Audioverarbeitung!

Im PRBSynth1 läuft dieser Software-Filter mit maximal 40 kHz\*. Die Variablen LP, BP, HP sind Integer-, die anderen 8-Bit-Variablen. Zwar erhält man so kein "HiFi-Filter",

aber da es sich bei dem Eingangssignal um Rauschen handelt, können wir einige Abstriche machen, solange es nicht zu zu hörbaren Artefakten kommt.

\*Ein Trick: Im Rauschmodus wird der Abstimmbereich des Filters um etwa eine Oktave erweitert, indem der Abtasttakt bei niedrigeren Filtereckfrequenzen schrittweise verringert wird. Bei Rauschen als Eingangssignal ist dies akzeptabel, wenn keine Aliasing-Probleme auftreten.



Potentiometer	CC code
Filter Eckfrequenz	1
Lautstärke	7
(immer maximieren)	
Panorama	10
Filtertyp und Güte	73
Rauschquelle	75
LFO Modulationstiefe	76
LFO Rate	77
LFO Wellenform	78
LFO Modus	79

Tabelle 1. Zuordnung von Potistellungen zu MIDI-CC-Code.

Eingangssignal durch die Nichtlinearität der Diodenübertragungskurven bei großen Signalen hinzugefügt wird.

Abschließend noch ein Hinweis zu R19. Dieser Widerstand kann eingebaut werden, wenn das Gerät von einer wiederaufladbaren 9-V-NiMH-Batterie (nicht Li-Po!) an JP3 versorgt wird, da er das Aufladen der Batterie über JP5 ermöglicht. In allen anderen Fällen ist sollte der Widerstand weggelassen werden.

## **Die Software**

Das Programm oder die Sound-Engine des PRBSynth1 ist recht einfach. Sie kann in vier parallel laufende Tasks unterteilt werden (Bild 4):

- > Audio-Sample-Verarbeitung
- > Modulation und Benutzeroberfläche
- > Panorama-Steuerung
- > MIDI-Verarbeitung

Im ersten und wichtigsten Task bestimmt Timer1 die Sample Rate. Bei jedem Tick erzeugt dieser Task ein Rauschsample, lässt es durch das Filter und den Verstärker laufen und sendet es dann an den DAC.

Der zweite Task lässt den LFO laufen, liest die Potentiometer und den Trigger-Taster aus und aktualisiert die modulierten Parameter. Er steht unter der Kontrolle von Timer0, der auch die LED-Intensitäten über Software-PWM steuert. Timer2 ist ein freilaufender Timer, der als PWM-DAC dient, um die Steuerspannung für den Stereo-Panner zu erzeugen. Dies ist der dritte Task.

Der vierte Task schließlich läuft im Hintergrund, wenn keine Interrupts bedient werden. Er verarbeitet eingehende MIDI-Daten. Dies ist möglich, weil MIDI relativ langsam ist (310 µs pro Nachricht) und der UART einen Puffer hat. Bei der Verwendung dieses Synthesizers mit einem Software-Sequenzer, der auf mehreren



## Erzeugung von Rauschen durch "Münzwurf"

Wenn man wiederholt (und mit hoher Geschwindigkeit) eine Münze wirft, erhält man einen Bitstrom, der sich wie ein zufälliges PWM-Signal verhält. Lässt man dieses Signal durch einen Tiefpassfilter laufen, erhält man einen Strom von gemittelten, zufällig im Bereich zwischen 0 und 1 schwankenden Werten.

Das Werfen einer Münze kann von einer Software simuliert werden, indem eine lange Folge von Zufallswerten erzeugt und dann nur ein Bit der Werte (zum Beispiel Bit 0) verwendet wird. Das Ergebnis ist zwar nicht mehr wirklich zufällig, aber es ist nahe genug daran. Eine solche Folge wird Pseudo-Zufallsbitfolge (PRBS, daher der Name des Synthesizers) genannt und kann von einem abgegriffenen Schieberegister mit EXOR-Rückkopplung erzeugt werden. Wenn die Sequenz lang genug ist (ein paar Sekunden oder länger), klingt sie wie Rauschen. Sie können ein solches abgegriffenes Schieberegister mit Rückkopplung

// init sr as word sr=1, init DAC as byte.

## **Rosa und rotes Rauschen**

Eine Besonderheit des PRBS-Signals besteht darin, dass bei der Abtastfrequenz F kein Signal im Spektrum vorhanden ist. Daher führt eine Verringerung des Abtasttakts nicht bei F, F/3, F/9 und F/27 und die Signale addieren. Dadurch wird der Anteil der niedrigen Frequenzen erhöht, der für die Erzeugung von rosa und rotem Rauschen erforderlich ist.

## Verringerung des Rechenaufwands

Die Erzeugung von rosa und rotem Rauschen auf diese Weise führt zu einer enormen Belastung des Controllers, wenn zu bestimmten Zeitpunkten zwei, drei oder sogar vier Münzen gleichzeitig geworfen werden müssen. Um dies zu verhindern, wird die langsamste Münze priorisiert: Nur die langsamste Münze wird geworfen, die anderen werden übersprungen. In der Praxis lassen sich keine hörbaren Unterschiede ausmachen.

## Muster für den Mustermodus erstellen

Das ist einfach, denn hier verkürzen wir einfach die Länge der Kette, indem wir das Schieberegister auf einen vom LFO aktualisierten Startwert zurücksetzen.



www.elektor.de/18615

www.elektor.de/17967







Bild 5. Der PRBSynth1, eingebaut in einem PVC-Brüstungskanal. Um die Platine klein und flexibel einbaubar zu halten, werden LEDs und Potentiometer über Stiftleisten verdrahtet. Eine 9-V-Batterie versorgt die Schaltung mit Strom.

MIDI-Kanälen aktiv ist, traten keine Probleme auf. Beachten Sie, dass der MIDI-Ausgang noch nicht im Code implementiert ist, aber die Hardware ist schon einmal vorhanden, falls für zukünftige Erweiterungen entsprechende Ideen auftauchen.

## Zusammengefasst

Der PRBSynth1 ist ein kleiner Synthesizer (**Bild 5**) mit vielen Möglichkeiten, auch wenn er "nur" auf einem Arduino Pro Mini läuft. Er kann auf einem Breadboard aufgebaut werden und bietet viel Raum für Hard- und Softwareexperimente. Alle Details zum Bau und zur Programmierung sowie Klangbeispiele finden sich auf der Webseite des Projekts bei Elektor Labs [1]. Ein Video ist auf YouTube [3] verfügbar. Viel Spaß beim Rauschen!

210003-03

## **Ein Beitrag von**

Idee und Entwurf: **Raymond Schouten** Redaktion: **Clemens Valens** Übersetzung: **Rolf Gerstendorf** Layout: **Harmen Heida** 

## 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 oder an Elektor unter redaktion@elektor.de.

## **MIDI-Implementierung**

Es wurden einige Anstrengungen unternommen, um die "Musikalität" des PRBSynth1 zu verbessern, wenn er an ein MIDI-Keyboard angeschlossen ist, vorzugsweise eines mit Velocity-, Pitch-Bend- sowie Modulationsrad-Controllern. Alle acht Potentiometer können mit einem MIDI-CC-Controller überschrieben werden. Durch Drehen des entsprechenden Potentiometers des Synthesizers wird die Kontrolle wieder zurückerlangt. Falls Sie eine reine MIDI-Version des Synthesizers planen, ist es daher notwendig, den Teil des Programms zu entfernen, der die Potentiometer ausliest. Wenn Sie das nicht tun, erzeugen die offenen Eingänge zufällig schwankende Werte und überschreiben die MIDI-Einstellung. In der Programmversion mit der Bezeichnung *no\_potm* ist dieser Teil bereits deaktiviert.

Wenn Sie ein MIDI-Keyboard (oder einen Sequenzer) verwenden, stellt das Modulationsrad die Filter-Eckfrequenz ein. Außerdem wird der Ton abgeschaltet, wenn keine Taste gedrückt wird. Dieser Modus wird beim Empfang der ersten MIDI-Note aktiviert und beim Einschalten zurückgesetzt.

Die Key Velocity (Anschlagsstärke) wird auf die Filtermodulationstiefe des LFOs abgebildet. Im *Noise Mode* hat der Sound keine Tonhöhe, daher steuert die Tastatur stattdessen die Geschwindigkeit des LFOs. Eine Oktave nach oben verdoppelt die Geschwindigkeit. Auf diese Weise können Sie leicht rhythmische Patterns erzeugen, wenn der LFO im *Continous Mode* ist. Wenn Sie den LFO in den *Trigger Mode* versetzen, können Sie lange/kurze Hüllkurvensounds durch das Spielen einer tiefen/hohen Taste auslösen.

Im *Pattern Mode* steuert die Tastatur die Länge des Patterns. Dies wurde so eingerichtet, um eine musikalische Tonleiter zu erzeugen (zum Beispiel halbiert eine Oktave nach oben die Länge) und ermöglicht das Spielen von Melodien. In diesem Modus wird die LFO-Geschwindigkeit mit dem entsprechenden Potentiometer eingestellt.

## \_ WEBLINKS \_

[1] PRBSynth1 bei Elektor Labs:

- www.elektormagazine.de/labs/synthesizer-prbsynth1
- [2] Website des Autors: http://www.rs-elc.nl
- [3] Video-Demonstration: https://youtu.be/agUzP0t1k7Y



ist gar nicht schwer! Weiter geht's mit der Spule... •

Von Eric Bogers (Elektor)



## Noch einmal: Die Spule als Wechselstromwiderstand

Genau wie ein Kondensator verhält sich auch eine Spule wie ein Wechselstromwiderstand - nur "andersherum". Um dies zur veranschaulichen, betrachten wir eine Reihenschaltung aus einer Spule und einem Widerstand.

Wir gehen aus von einer Spule mit einer Eigeninduktivität von 10 mH und einem Widerstand von 50  $\Omega$  bei einer Frequenz von 1 kHz. Bei dieser Frequenz beträgt der Blindwiderstand der Spule:

$$X_{\scriptscriptstyle L} = 2 \cdot \pi \cdot f \cdot L = 2 \cdot \pi \cdot 1000 \; \mathrm{Hz} \cdot 0,01 \; \mathrm{H} = 62,8 \; \Omega$$

Auch jetzt dürfen wir diese Impedanz nicht einfach zum ohmschen Widerstand addieren. Dies muss vektoriell geschehen:

$$X_{tot} = \sqrt{X_L^2 + R^2} = \sqrt{(68, 2\Omega)^2 + (50\Omega)^2} = 80, 3\Omega$$

Zwischen dem Strom und der Spannung gibt es (genau wie bei einem Kondensator) eine Phasenverschiebung. Die Phasenverschiebung beträgt hier:

$$\varphi = \arctan \frac{X_L}{R} = \arctan \frac{62,8\Omega}{50\Omega} = 51,5^{\circ}$$

Wir wollen Sie nicht mit der Berechnung für die Parallelschaltung von Spule und Widerstand belästigen; mit denselben Bauteilwerten erhalten wir eine Gesamtimpedanz von 39,1  $\Omega$  mit einem Phasenwinkel von -38,5° (überprüfen Sie es übungshalber selbst).

## **Hoch- und Tiefpassfilter**

Hoch- und Tiefpassfilter können nicht nur aus Kondensatoren und Widerständen, sondern auch aus Induktivitäten und Widerständen aufgebaut werden (**Bild 1**).

Für den Frequenz- und Phasenverlauf gelten die gleichen Überlegungen wie bei Filtern aus Kondensatoren und Widerständen. Bei der Eckfrequenz ist die Impedanz gleich dem Widerstand:

$$R = 2 \cdot \pi \cdot f \cdot L \rightarrow f_{Eck} = \frac{R}{2 \cdot \pi \cdot L}$$

Es ist eigentlich ganz naheliegend, Induktivitäten und Kondensatoren in Filterschaltungen zu kombinieren. Und genau das werden wir später auch tun. Doch zunächst etwas anderes.

## **Blindleistung und ihre Kompensation**

Eine fantastische Anwendung ist die Kompensation von Blindleistung. Dazu müssen wir aber erst einmal wissen, was Blindleistung eigentlich ist.

Die Leistung, die in einem ohmschen ("normalen") Widerstand verloren geht, wird von diesem Widerstand in Wärme umgewandelt. Die von einem Kondensator oder einer Induktivität aufgenommene Leistung hingegen wird in diesen Bauteilen in Form einer elektrischen Ladung oder eines Magnetfeldes gespeichert. Diese gespeicherte Energie wird dann bei der ersten Gelegenheit wieder abgegeben. Über einen längeren Zeitraum betrachtet, verbrauchen Kondensatoren und Induktivitäten also überhaupt keinen Strom. In einer Formel ausgedrückt (die Sie, was uns betrifft, sofort vergessen können):

$$p = \int_{0}^{2\pi} U_0 \cdot \sin \varphi \cdot I_0 \cdot \cos \varphi = U_0 \cdot I_0 \cdot \sin^2 \varphi \Big|_{0}^{2\pi} = 0$$
$$p = \int_{0}^{2\pi} U_0 \cdot \sin \varphi \cdot I_0 \cdot -\cos \varphi = U_0 \cdot I_0 \cdot -\sin^2 \varphi \Big|_{0}^{2\pi} = 0$$

Der Grund dafür ist die Tatsache, dass Strom und Spannung nicht in Phase sind, sondern genau um 90° zueinander verschoben sind. Das bedeutet, dass die Leistung mal positiv und dann wieder negativ ist, aber integriert über eine ganze Periode ist sie gleich Null. In einem ohmschen Widerstand hingegen sind Strom und Spannung immer gleichphasig, so dass die Leistung immer positiv ist. Die elektrische Leistung, die von einem ohmschen Widerstand aufgenommen wird, wird in anderer Form wieder abgegeben - zum Beispiel in Form von Wärme - und man spricht in diesem Zusammenhang von *Wirkleistung*. Das, was Induktoren und Kondensatoren vorübergehend aufnehmen, nennt man dagegen *imaginäre Leistung* oder *Blindleistung*. In Formeln wird dafür das Symbol Q verwendet. Eine Schaltung, die neben ohmschen Elementen auch Kondensatoren und (oder Induktivitäten enthält, uerbraucht" gewehl zoale

toren und/oder Induktivitäten enthält, "verbraucht" sowohl reale als auch blinde Leistung. Da diese Leistungskomponenten nicht die gleiche Phase haben, dürfen wir sie nicht einfach addieren. Stattdessen gilt die folgende Formel (**Bild 2**).

$$S = \sqrt{p^2 + Q^2}$$



Bild 1. RL-Filter.

Hier ist S die Scheinleistung. Um sie von der Wirkleistung zu unterscheiden, wird S nicht in Watt (W), sondern in Voltampere (VA) angegeben. Für die Wirkleistung und die Imaginärleistung kann man nun schreiben:

$$P = S \cdot \cos \varphi \qquad \qquad Q = S \cdot \sin \varphi$$

Auf dem Typenschild eines Elektromotors ist häufig der Wert  $\cos \varphi$ angegeben, so dass die Scheinleistung aus der Wirkleistung abgeleitet werden kann. Nehmen wir an, dass für einen bestimmten Motor eine Spannung von 230 V, ein Strom von 2 A und ein  $\cos \varphi$  von 0,8 angegeben ist. Für die Wirkleistung ergibt sich dann

$$P = U \cdot I \cdot \cos \varphi = 230 \text{ V} \cdot 2 \text{ A} \cdot 0, 8 = 368 \text{ W}$$

Und da das Folgende wahr ist

$$\sin^2 \varphi + \cos^2 \varphi = 1 \rightarrow \sin \varphi = \sqrt{1 - \cos^2 \varphi} = \sqrt{1 - (0, 8)^2} = 0, 6$$

finden wir für die Blindleistung:

$$Q = U \cdot I \cdot \sin \varphi = 230 \text{ V} \cdot 2 \text{ A} \cdot 0, 6 = 276 \text{ VA}$$

Nun mögen Sie als Nutzer dieses Motors vielleicht denken, dass die Blindleistung für Sie völlig unbedeutend ist - schließlich misst der Stromzähler im Zählerkasten nur die Wirkleistung. Aber als Folge dieser imaginären Leistung fließt ein sinnloser Strom in den Stromkabeln, die dann entsprechend größer dimensioniert werden müssen; außerdem verursacht dieser "sinnlose" Strom reale Verluste im Kabelwiderstand. Grund genug also, die Blindleistung zu kompensieren. Dazu brauchen wir nur einen entsprechend dimensionierten Kondensator parallel oder in Reihe mit der Last zu schalten (**Bild 3**).

Betrachten wir zunächst die Parallelkompensation (die übrigens



Bild 2. Scheinleistung, Wirkleistung und Blindleistung.



Bild 3. Parallelkompensation (links) und Reihenkompensation (rechts).

in der Praxis bevorzugt wird). Die Blindleistung ist vollständig kompensiert, wenn die Blindleistung des Kondensators gleich der der Spule ist. In diesem Fall gibt der Kondensator seine imaginäre Leistung an die Spule ab und umgekehrt, und die gesamte Schaltung verhält sich (von außen betrachtet) wie eine ohmsche Last (bei der betreffenden Frequenz, versteht sich). Für den Strom durch den Kondensator gilt:

$$I = \frac{Q_c}{U} = \frac{276 \text{ VA}}{230 \text{ V}} = 1,2 \text{ A}$$

Für die Impedanz bedeutet dies:

$$X_{C} = \frac{U}{I} = \frac{230 \text{ V}}{1,2 \text{ A}} = 191,67 \Omega$$

Und damit für die Kapazität:

$$C = \frac{1}{2 \cdot \pi \cdot f \cdot X_c} = \frac{1}{2 \cdot \pi \cdot 50 \text{ Hz} \cdot 191,67 \Omega} = 16,6 \,\mu\text{F}$$

Durch diese Maßnahme verringert sich der Gesamtstrom von 2 A auf 1.6 A. während sich der Gesamtwiderstand des Stromkreises von 115  $\Omega$  auf 143,75  $\Omega$  erhöht. Der Gesamtwiderstand erhöht sich? Sicherlich ist dies auf die entgegengesetzten Phasenwinkel des Kondensators und der Spule zurückzuführen.

Wir haben festgestellt, dass eine Parallelkompensation im Allgemeinen vorzuziehen ist. Wenn wir im obigen Beispiel eine Reihenkompensation verwenden würden, müsste der Kondensator einen Wert von 46,1 µF haben - das ist fast dreimal so groß. Die Parallelkompensation ist daher in der Praxis (preis-) günstiger zu realisieren. Außerdem steigt bei der Reihenkompensation der Gesamtstrom an, anstatt zu sinken. 

210374-0

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 Kommentare zu diesem Artikel? Senden Sie eine E-Mail an den Autor oder an die Elektor-Redaktion über redaktion@elektor.de.

## **Ein Beitrag von**

Idee und Illustrationen: Michael Ebner Text und Redaktion: Eric Bogers Übersetzung: Rolf Gerstendorf Layout: Giel Dols

## PASSENDE PRODUKTE

- > Ebner, Michael, "Basiskurs Elektronik" www.elektor.de/basiskurs-elektronik-pdf
- > B. Kainka, Elektronik-Grundlagen und Einsteiger-Projekte (E-Book), Elektor 2020 www.elektor.de/ elektronik-grundlagen-und-einsteiger-projekte-pdf



## Die Neuronen in neuronalen Netzen verstehen Teil 2: Logische Neuronen

## Von Stuart Cording (Elektor)

Im ersten Teil dieser Artikelserie haben wir erfahren, wie Forscher sich langsam der Funktionalität des Neurons angenähert haben. Der wirkliche Durchbruch bei künstlichen Neuronen kam mit dem mehrschichtigen Perzeptron (MLP) und der Verwendung der Backpropagation, um ihm beizubringen, wie man Eingaben klassifiziert. Anhand einer selbst erstellten Implementierung eines MLP in Processing haben wir außerdem gezeigt, wie es funktioniert und seine Gewichte zum Lernen anpasst. Nun greifen wir auf die Experimente der Vergangenheit zurück, um unserem neuronalen Netz beizubringen, wie Logikgatter funktionieren und prüfen, ob unser MLP in der Lage ist, die XOR-Funktion zu lernen.

Wir haben eine flexible Neural-Klasse zur Implementierung eines MLP, die in ein Processing-Projekt eingebunden werden kann. Aber die bisherigen Beispiele bestätigen lediglich die korrekte Berechnung des Vorwärtsdurchlaufs und wie die Backpropagation die Gewichte des Netzwerks anpasst, um eine bestimmte Aufgabe zu erlernen. Nun können wir dieses Wissen an einer echten Aufgabe erproben, die schon in frühen Betrachtungen der McCulloch-Pitts Threshold Logic Units (TLU) gestellt wurde: die Implementierung von Logik. Wie wir bereits festgestellt haben, sollte unser MLP linear trennbare Probleme wie UND und ODER mit Leichtigkeit lösen. Aber es sollte auch in der Lage sein, die XOR-Funktion zu lösen, etwas, was die TLU und andere frühe künstliche Neuronen nicht konnten. Wir werden wir auch untersuchen, wie diese Netzwerke dank einer visuellen Implementierung des neuronalen Netzwerks lernen und welche Auswirkung die gewählte Lernrate auf den Ausgabefehler hat.

## UND

Ein neuronales Netzwerk kann zwar lernen, die UND-Verknüpfung nachzubilden, aber es funktioniert nicht auf dieselbe Weise. Wir wenden statt dessen Eingaben in ein Netzwerk an, das die UND-Funktion gelernt hat, und fragen es: "Wie sicher bist du, dass diese Eingabekombination das Muster ist, dem wir eine 1 zuschreiben?" Das Beispielprojekt im GitHub-Repository im Ordner /processing/and/and.pde demonstriert dies, wenn es mit Processing geöffnet wurde. Unser neuronales Netzwerk hat wie ein UND-Gatter zwei Eingänge und einen Ausgang. Zwischen den Eingangs- und dem Ausgangsknoten sind vier versteckte Knoten (**Bild 1**). Wie man die Anzahl der benötigten



Bild 1. MLP-Konfiguration zum Lernen der UND-Funktion (Biases werden nicht gezeigt).

versteckten Knoten bestimmt, werden wir später besprechen. Der Code zur Vorbereitung des Netzes ist in **Listing 1** dargestellt. Das Netz soll so trainiert werden, dass es das Muster "11" an den Eingängen erkennt. Außerdem wollen wir sicherstellen, dass die Alternativen "00", "01" und "10" die Klassifizierungsschwelle nicht überschreiten. Beim Training des Netzwerks werden die Eingangswerte und erwarteten Ergebnisse aus **Tabelle 1** angewendet.

Eing	ang	Erwarteter Ausgang
i <sub>1</sub>	i <sub>2</sub>	0 <sub>1</sub>
0,01	0,01	0,01
0,01	0,99	0,01
0,99	0,01	0,01
0,99	0,99	0,99

Tabelle 1. Eingaben und erwartete Ausgabe des MLP, wenn "UND" erfolgreich erlernt wurde.

Beachten Sie, dass solche Netze nicht mit logischen, sondern mit dezimalen Werten arbeiten. Eine "1" entspricht hier 0,99 (fast 1), eine 0 dem Dezimalwert 0,01 (fast 0). Die Ausgabe wird dann ebenfalls zwischen 0,0 und 1,0 liegen. Das Ergenis sollte deshalb auch als Zuverlässigkeit betrachtet werden, dass die Eingaben mit

der gelernten Klassifizierung übereinstimmen, zum Beispiel 96,7 % Wahrscheinlichkeit, dass beide Eingaben "1" sind. Das entspricht nicht der eindeutigen 0/1-Ausgabe eines echten UND-Gatters! Wir können den Lernerfolg des neu erstellten Netzwerks überprüfen, indem wir ihm einige Eingabewerte übergeben und die Ausgabe abfragen. Da der Konstruktor den Gewichtungen Zufallswerte zuweist, dürften sich erzeugten Ergebnisse jedes Mal unterscheiden. Der folgende Code gibt die Ausgabe des Netzwerks auf die Eingangswerte "11" und "00" aus. Es ist sehr wahrscheinlich, dass das Ergebnis für "11" nahe bei 0,99 liegt, während das Ergebnis für "00" viel größer als die erhofften 0,01 sein dürfte:

## // Check output of AND function for 00 input

```
network.setInputNode(0, 0.01);
network.setInputNode(1, 0.01);
network.calculateOutput();
println("For 00 input, output is: ",
network.getOutputNode(0));
// Check output of AND function
for 11 input
network.setInputNode(0, 0.99);
network.setInputNode(1, 0.99);
network.calculateOutput();
println("For 11 input, output is: ",
network.getOutputNode(0));
```

## -----

Listing 1. Abschnitt von and.pde, der das neuronale Netz zum Lernen der UND-Funktion konfiguriert.

// We'll use two inputs, four hidden nodes, and one output node.
network = new Neural(2,4,1);

```
// Set learning rate here
network.setLearningRate(0.5);
```

println(network.getNoOfInputNodes(), " ", network. getNoOfHiddenNodes(), " ", network.getNoOfOutputNodes());

// Set network biasing here
network.setBiasInputToHidden(0.25);
network.setBiasHiddenToOutput(0.3);

network.displayOutputNodes();

println(network.getTotalNetworkError());

network.turnLearningOn();

Dies lieferte im Test folgende Ausgaben:

## For 00 input, output is: 0.7490413 For 11 input, output is: 0.80063045

Wir konstatieren, dass das neuronale Netz bei 0,99 beider Eingänge denkt, dass die Eingabe "1 UND 1" ist, und dies mit einer Zuverlässigkeit von 0,8006 oder 80,06 %. Dies ist in diesem frühen Stadium nicht schlecht. Wenn jedoch beide Eingangswerte 0,01 sind, ermittelt das neuronale Netzwerk eine Wahrscheinlichkeit von 0,7490 (74,90 %), dass die Eingabe "1 UND 1" ist. Das ist leider etwas weit von den erhofften 0 % entfernt. Um dem Netzwerk beizubringen, wie eine UND-Verknüpfung funktioniert, muss es trainiert werden. Dazu werden die Eingänge und der gewünschte Ausgang für alle vier Fälle (Eingänge 00, 01, 10 und 11, Ausgang 0, 0, 0 und 1) entsprechend eingestellt und nach jeder Änderung die Methode calculateOuput() mit aktiviertem "learning" in einer Schleife aufgerufen:

while (/\* learning the AND function \*/) { // Learn 0 AND 0 = 0 network.setInputNode(0, 0.01); network.setInputNode(1, 0.01); network.setOutputNodeDesired(0, 0.01); network.calculateOutput(); // Learn 0 AND 1 = 0 . . . // Learn 1 AND 0 = 0 . . . // Learn 1 AND 1 = 1 network.setInputNode(0, 0.99); network.setInputNode(1, 0.99); network.setOutputNodeDesired(0, 0.99); network.calculateOutput(); } network.turnLearningOff();

Die Entscheidung, das Training des Netzwerks zu beenden, kann auf verschiedene Weisen erfolgen. Jeder Lernzyklus wird in diesem Beispiel als eine "Epoche" bezeichnet. Der Lernprozess kann nach einer bestimmten Anzahl von Epochen, etwa 10.000, gestoppt werden. Alternativ dazu kann auch der Ausgabefehler als Kriterium hinzugezogen werden. Sobald er beispielsweise unter 0,01 % liegt, kann das Netzwerk als genau genug angesehen werden.

Sie sollten beachten, dass ein MLP nicht immer zu dem gewünschten Ergebnis führt. Sie könnten aufgrund der gewählten Kombination von Eingangs- und Bias-Gewichtungen kein Glück haben, andererseits könnte es passieren, dass die Konfiguration des MLP aufgrund von zu vielen oder zu wenigen versteckten Knoten nicht in der Lage ist, Ihre Aufgabe zu lernen. Hier gibt es keine Regeln – die geeignete Anzahl von Knoten, Eingangs- und Biasgewichtungen kann nur durch Versuch und Irrtum oder Erfahrung ermittelt werden. Für dieses Beispiel haben wir vier versteckte Knoten gewählt, da vier Eingangszustände gelernt werden müssen. Dabei bleibt es zu hoffen, dass jeder versteckte Knoten einen Zustand erlernt.

Außerdem sollte das Training stufenweise erfolgen, das heißt, der Trainingsdatensatz sollte wiederholt von Anfang bis Ende durchlaufen werden. Wenn "O UND O = o" mehrere tausend Zyklen gelehrt wird, tendiert das Netzwerk zu diesem Ergebnis, und es wird dann fast unmöglich, die verbleibenden Eingangsdaten zu trainieren.

## Visualisierung

Da die grundlegende Implementierung nun bekannt ist, können wir uns das Beispiel, in dem das Netzwerk die UND-Funktion lernt, genauer ansehen. Zum besseren Verständnis, wie neuronale Netze lernen, wird das Netz in der Anwendung während des Lernprozesses und später im Betrieb visualisiert.

Wenn Sie auf Run klicken, sollten Sie eine Ausgabe wie im Screenshot in **Bild 2** erhalten. Zu Beginn befindet sich die Anwendung im Lernmodus und bringt dem Netz die erwartete Ausgabe für eine UND-Funktion für die beiden Eingangswerte (links) bei. Während des Lernvorgangs wechseln die Eingangswerte schnell zwischen den logischen Werten 0 und 1. Auf der rechten Seite befindet sich der Ausgangsknoten, der anfangs 0 ist. Die Entscheidung, eine 1 auszugeben, wird nur dann getroffen, wenn eine Zuverlässigkeit von >90% gegeben ist, dass beide Eingänge 1 sind. Andernfalls ist der Ausgangswert O. Diese Entscheidung wird zwischen den Zeilen 341 und 346 in and.pde getroffen.

// Output Node Text
if (network.getOutputNode(0) > 0.9) {
text("1", 550, 280);
} else {
text("0", 550, 280);
}

Anfänglich bleibt die Ausgabe 0, da die Zuverlässigkeit 90 % noch nicht erreicht hat. Nach ungefähr 5.000 Epochen sollte der Ausgabewert beginnen, zwischen o und 1 hin und her zu flattern, was zeigt, dass er begonnen hat, erfolgreich zu erkennen, dass die Eingangswerte 11 eine 1 ausgeben sollten. Zu diesem Zeitpunkt liegt der Gesamtfehler des Netzwerks bei etwa 0,15 %. Wenn dies nicht geschieht, ist der Lernprozess wahrscheinlich steckengeblieben.

Während des Lernprozesses werden in Processing die Gewichtungen zwischen den Knoten als Linien mit unterschiedlicher Dicke und Farbe angezeigt. Je dicker die Linie ist, desto größer ist der Wert. Schwarze Linien zeigen positive und braune Linien negative Zahlen.

Bei jeder Ausführung des Codes ändern sich die Linien, doch allmählich entwickelt sich ein Muster. Zwei versteckte Knoten haben immer eine braune und eine schwarze Linie, ein versteckter Knoten hat zwei schwarze Linien und einer hat zwei braune Linien. Die Linien zwischen den versteckten Knoten und dem Ausgangsknoten entwickeln ebenfalls ein Muster: Die einzige schwarze Linie geht von dem Knoten mit zwei eingehenden schwarzen Gewichtungslinien aus.

Dies ist eine interessante Erkenntnis, da sie zeigt, wie das Netzwerk die UND-Funktion gelernt hat. Eine oo am Eingang lässt sich leicht in eine o am Ausgang überführen, ebenso eine 11 in eine 1. Für die Kombinationen 01 und 10 scheint es, dass hauptsächlich die Nullen dafür verantwortlich sind, dass die Ausgabe in Richtung o gedrückt wird.

Die Anwendung ist so programmiert, dass sie aufhört zu lernen, sobald in Zeile 57 von and.pde der Gesamtfehler des Netzwerks <0,05 % beträgt. Wahlweise kann der Lernvorgang auch so programmiert werden, dass er (in Zeile 55) nach einer bestimmten Anzahl von Epochen stoppt. Nach Abschluss des Lernvorgangs durchläuft die Anwendung einfach der Reihe nach die binären Eingangskombinationen, um zu zeigen, was das Netz gelernt hat (**Bild 3**). In der Textkonsole werden die Eingaben (als Dezimalwerte 0...3) mit der berechneten Ausgabe im Textformat angezeigt:

0 : 5.2220514E-4 1 : 0.038120847 2 : 0.04245576

```
3 : 0.94188505
```
0 : 5.2220514E-4 1 : 0.038120847 2 : 0.04245576 3 : 0.94188505

Für Interessenten werden der Ausgabefehler für die angewendeten Eingangswerte und der durchschnittliche Netzwerkfehler alle 50 Epochen in eine CSV-Datei namens *and-error.csv* geschrieben. Die Datei kann in Excel importiert werden, um zu überprüfen, wie das Netzwerk auf den Lernerfolg zustrebt (**Bild 4**). Die Grafik zeigt, wie der Fehler zunächst zwischen hohen und niedrigen Werten für bestimmte Eingabe/Ausgabe-Kombinationen hin und her schwankt. Der hohe Fehler liegt wahrscheinlich darin begründet, dass die Ausgabe für die Muster 00, 01 und 10 in der frühen Phase des Lernens viel zu hoch war. Der niedrige Fehler wird verursacht, wenn das Netz die Eingabe 11 schon früh korrekt auswertet. Die individuellen (momentanen) Fehler hier werden über vier Epochen gemittelt, um den durchschnittlichen Fehler zu berechnen. Sollte Ihr PC kein "Englisch" können, müssen Sie vor dem Export in Excel mit einem Texteditor wie Notepad++ die Trennzeichen-Kommas in der CSV-Datei durch Semikolons und die Punkte durch Kommas ersetzen.

Mit der CSV-Datei kann man auch die Auswirkungen der Lernrate auf das Netzwerk zu untersuchen. Der Beispielcode verwendet eine Lernrate  $\eta$  von 0,5. Höhere Lernraten führen zwar dazu, dass das Netz schneller lernt, wie in **Bild 5** zu sehen ist, aber es besteht auch die Gefahr von Oszillationen, so dass das Netz nie das gewünschte



Bild 2. Screenshot des Neuronalen Netzes während der Lernphase der UND-Funktion.



Bild 3. Sobald das UND-Muster gelernt wurde, durchläuft die Anwendung die vier binären Eingangskombinationen und zeigt die Ausgangswerte des Netzwerks an.



Bild 4. Alle 50 Epochen des Lernens von UND werden momentaner und durchschnittlicher Fehler ermittelt.



Bild 5. Einfluss der Lernrate n auf den Fehler beim Lernen (erste 10.000 Epochen).



Bild 6. Das MLP während des Lernprozesses von XOR zeigt, dass es Schwierigkeiten hat, geeignete Gewichte zu finden.

Lernergebnis erzielen kann. Doch alle hier getesteten Lernraten führten zu einem korrekt funktionierenden Netzwerk, das UND gelernt hatte. Es ist jedoch zu beachten, dass die Startgewichtungen jedes Mal zufällig gewählt wurden.

## Kann das Netz auch XOR lernen?

Das Repository enthält in processing/or/ or.pde auch Beispiele für eine ODER-Funktion. Da diese Aufgabe linear trennbar ist, hat der MLP auch keine Probleme, diese Funktion zu erlernen. Aber schauen Sie es sich ruhig den Unterschied in den Gewichtungen nach dem Lernen im Vergleich zum UND-Beispiel an! Sowohl or.pde als auch and.pde können leicht geändert werden,



#### **PASSENDE PRODUKTE**

- B. van Dam, Artificial Intelligence (E-Buch) (SKU 18090) www.elektor.com/artificial-intelligence-e-book
- Google AIY Vision Kit für Raspberry Pi (SKU 1936 www.elektor.de/google-aiy-vision-kit-for-raspberry-pi
- HuskyLens AI-Kamera mit Silikon-Gehäuse (SKU 19248 www.elektor.de/huskylens-ai-camera-with-silicone-case

um dem Netzwerk die Nicht-UND- und Nicht-ODER-Funktionen beizubringen. Der Moment der Wahrheit kommt jedoch mit der XOR-Funktion.

Ein Beispiel hierfür ist in processing/xor/ xor.pde zu finden. Der Code ist ähnlich aufgebaut und verwendet die gleiche 2/4/1-MLP-Knotenkonfiguration (Eingabe/ versteckt/Ausgabe) (**Bild 6**). Bei der verwendeten Lernrate ( $\eta = 0.5$ ) werden wahrscheinlich 15.000 Epochen oder mehr benötigt, bevor sich die Ausgabe zu ändern beginnt, und etwa 35.000 Epochen, bis der angestrebte durchschnittliche Fehler von 0,05 % erreicht ist.

Es ist klar, dass das Netzwerk Schwierigkeiten hat, die XOR-Funktion zu erlernen. Dies spiegelt sich in den angezeigten Gewichten



Bild 7. Die Herausforderungen beim Erlernen von XOR spiegeln sich in der Fehlerausgabe wider.

wider, die zwischen positiv und negativ hin und her springen, bevor sie eine Richtung wählen, und der Netzwerkfehler nimmt sehr langsam ab. Das liegt daran, dass sowohl oo und 11 (dargestellt als 0,01/0,01 sowie 0,99/0,99 an den Eingängen) die Ausgabe 0,01 liefern sollen. Mathematisch gesehen führen Eingangswerte von 0,99 zu hohen Ausgangswerten, bis das Netzwerk in der Lage ist, das Ergebnis während des Lernens in Richtung 0,01 zu drücken. Dies ist im Ausgabefehler zu sehen, der während des Lernvorgangs in *xor-error.csv* gespeichert wird (**Bild 7**).

Trotz der Herausforderungen bei dieser Aufgabe erlernt das Netzwerk letztendlich die XOR-Funktion. Sobald der Fehler unter 0,05 % liegt, wendet der Code die binären Eingaben auf das Netzwerk an, und die Ausgabe reagiert, indem sie die Muster 01 und 10 korrekt erkennt. Der Code zeigt dann eine 1 am Ausgangsknoten an (Bild 8). Wie beim UND-Code können wir sehen, wie das Netzwerk die XOR-Funktion gelernt hat. Zwei versteckte Knoten (die beiden mittleren in Bild 8) haben eine schwarze und eine braune Linie, die in sie hineinführen, und eine dicke schwarze Linie, die sie verlässt. Diese scheinen für die Klasifizierung von 01 und 10 verantwortlich zu sein. Das Netzwerk löste auch 00 an den Eingängen in O am Ausgang recht gut auf (oberer versteckter Knoten). Der Fall 11 scheint vom unteren versteckten Knoten verarbeitet zu werden, aber dies wurde möglicherweise während des Lernprozesses nicht sehr gut aufgelöst, was zu einem (noch unerwünscht) höheren Fehler als für die Eingabe 11 führte. Ein erneutes Ausführen des Codes würde wahrscheinlich dazu führen, dass ein versteckter Knoten

offensichtlich die 11 verarbeitet und zwei schwarze Linien in einen der versteckten Knoten eintreten (**Bild 9**).

#### **Beim nächsten Mal**

Eines der wichtigsten Dinge, die Sie mitnehmen sollten, dürfte die Erkenntnis sein, dass es bei neuronalen Netzen keine richtige oder falsche Antwort gibt. Das Netzwerk selbst klassifiziert lediglich, wie wahrscheinlich es ist, dass die Eingaben, die Sie bereitgestellt haben, die gesuchten Eingaben sind. Wenn Sie das Netz konfiguriert und trainiert haben und es das gewünschte Ergebnis liefert, ist es wahrscheinlich richtig. Idealerweise möchten Sie dies mit einer minimalen Anzahl von Knoten erreichen, um Speicher und Rechenzeit zu sparen.

Eine Visualisierung ist zwar nett, aber nicht unbedingt notwendig. Wenn Sie daran interessiert sind, mehr herauszufinden, können Sie im Projekt processing/fsxor/ fsxor.pde die CSV-Visualisierungen entfernen. Dies beschleunigt den Code erheblich. Sie können dann Ihren eigenen Code in der Klasse Neural schreiben, um Folgendes zu untersuchen:

- > Welchen Einfluss hat die Lernrate auf das Netzwerk beim Lernen von XOR? Untersuchen Sie es vielleicht jedes Mal mit den gleichen Startgewichtungen.
- > Können Sie die Gewichte mit Werten initialisieren, die das Netz dazu in weniger Epochen das Problem lösen zu lassen? Überprüfen Sie dies vielleicht anhand der Ausgangsgewichte aus einem früheren Durchlauf.
- > Wie wenige versteckte Knoten brauchen Sie, um UND zu lernen? Und wie wenige, um XOR zu lernen? Können Sie zu viele versteckte Knoten haben?
- > Ist es sinnvoll, zwei Ausgangsknoten zu haben? Einer könnte die unerwünschten Muster als 0,99 klassifizieren (für XOR oo und 11), während der zweite die

gewünschten Muster als 0,99 klassifiziert (für XOR 10 und 01).

Im nächsten Artikel über neuronale Netze werden wir das neuronale Netz darauf trainieren, Farben von einer an unseren PC angeschlossenen Webcam zu erkennen. Wenn Sie möchten, können Sie eine MLP-Knotenkonfiguration entwickeln und testen, von der Sie glauben, dass sie der Aufgabe gewachsen ist.

210160-B-02

#### Haben Sie Fragen oder Kommentare ?

Haben Sie Fragen oder Kommentare zu diesem Artikel? Dann schreiben Sie bitte dem Autor eine E-Mail an stuart.cording@ elektor.com.

#### Ein Beitrag von

Idee, Text und Bilder: **Stuart Cording** Redaktion: **Jens Nickel**, **C. J. Abate** Illustrationen: **Patrick Wielders** Übersetzung: **Textmaster** Layout: **Harmen Heida** 

#### **Kreatives Lernen**

KI und ML werden oft verwendet, um Bilder und komplexe Datensätze clever zu klassifizieren. Aber kann das auch kreativ sein? Das FlowMachines-Projekt des SONY CSL Research Laboratory möchte Sie das glauben machen. Mit "Daddy's Car" hat ihre KI scheinbar einen Beatles-ähnlichen Hit geschrieben [4]. Wenn man jedoch etwas tiefer gräbt, stellt sich heraus, dass viel menschlicher Input nötig war, um den Song zu arrangieren, zu produzieren und den Text zu schreiben. Kreativ oder nicht, der Prozess ist innovativ und aufregend, und eine solche KI könnte ein wertvoller Sparringspartner sein, wenn die menschlichen kreativen Säfte plötzlich versiegen.





Bild 8. Das Netzwerk zeigt, dass es die Funktionalität von XOR erfolgreich erlernt hat.



Bild 9. Nach dem erneuten Ausführen des XOR-Codes ist der obere versteckte Knoten offensichtlich für die Klassifizierung 11 zuständig, während der dritte versteckte Knoten für 00 zuständig ist.

#### **WEBLINKS**

- [1] GitHub Repository simple-neural-network: http://bit.ly/2ZHLv9p
- [2] Processing-IDE: https://processing.org/
- [3] J. Brownlee, "How to Configure the Number of Layers and Nodes in a Neural Network", Juli 2018: http://bit.ly/3aMHqam
- [4] T. Lee, "Daddy's Car' Is A Pop Song Composed By Artificial Intelligence", Übergizmo, September 2016: http://bit.ly/3shyfo6

# Probleme mit der Sicherheit? Bekämpfen Sie Feuer mit Feuer!

Blitzlicht-geschützter Analogspeicher-Zusatz für das Manipulationssichere Datenpaket

Von Luka Matić

Im vorigen Artikel unserer Projektreihe über hochsichere DIY-Kommunikation [1] wurde gezeigt, wie man eine sichere manipulationssichere Box baut, die den Empfänger Bob warnt, dass sie geöffnet wurde, nachdem Alice sie zur Post gebracht hatte. In diesem Artikel werden wir diese Box mit einer Technologie vom Anfang des letzten Jahrhunderts noch sicherer machen.

Die Sicherheit des Manipulationssicheren Datenpakets basiert auf dem Löschen der Challenge- und Response-Codes im SRAM des Mikrocontrollers bei unbefugtem Öffnen des Pakets. Da die Spionin Eve die Codes nicht aus dem SRAM auslesen kann, kann sie das Manipulationssichere Datenpaket nicht in seinen ursprünglichen Zustand zurückversetzen, und Bob wird gewarnt. Die Kommunikation zwischen Alice und Bob sollte damit nun völlig sicher sein, oder etwa nicht?

Eve ist zwar nicht in der Lage, den ursprünglichen Zustand wiederherzustellen, aber was ist, wenn Mallory vorher eingreift? Alles, was sie tun muss, ist, den Mikrocontroller, den Alice bei ihrem Lieblingslieferanten bestellt hat, durch eine vorbereitete Spionage-Variante zu ersetzen. Solange Alice keinen Zugang zu High-End-Laborgeräten hat, kann sie nicht feststellen, ob es sich bei ihrer MCU um ein jungfräuliches Original oder eine modifizierte bösartige Variante handelt. Mallory kann Hardware-Backdoors einbauen, ohne die Baupläne der Mikroschaltungen auf dem Siliziumchip [2] zu verändern, indem sie "einfach" die Dotierung manipuliert. Die Mikroschaltungen sehen dann selbst unter Alices Super-Duper-Mikroskop unverändert aus!

Das Öffnen der manipulationssicheren Box kann zuverlässig erkannt werden, aber die Nullsetzung des SRAM ist möglicherweise nicht zu 100 % wirksam. Auch wenn die Speicherbytes, die die kritischen Codes enthalten, regelmäßig bitweise invertiert werden, um verbleibende langfristige Einbrenneffekte zu begrenzen, kann Eve immer noch versuchen, die verbleibende SRAM-Remanenz - zumindest theoretisch – für einem so genannten Cold-Boot-Angriff [3] zu nutzen und den Inhalt zu lesen. Bedeutet dies, dass Alice und Bob die Kryptografie zu Heimwerker-Kosten aufgeben und einen Arbeitsplatz bei einer großen Agentur mit drei Buchstaben anstreben sollten?

Wenn die beiden aber das Manipulationssichere Datenpaket um einen wirklich nicht wiederherstellbaren Speicher erweitern, können sie dennoch ihre Low-Budget-Guerilla-Operationen fortsetzen. Dies funktioniert sowohl gegen allgemeine als auch anwendungsspezifische Hardware-Trojaner. Nach der Lektüre von [4], [5] und [3] wird Ihnen jedoch schnell klar, dass jede Form von digitalem Speicher, die der Menschheit bekannt ist, unter einer Art von unerwünschter Datenremanenz leidet. Wenn es also nicht digital geht, dann eben analog!

#### Flash-Speicher der besonderen Art

Magnesium-Blitzlampen **(Bild 1)** werden in der Fotografie fast seit ihrer Erfindung zur künstlichen Beleuchtung verwendet. Einwegblitzlampen wie der Standardtyp AG-1B nutzen die sehr hohe Temperatur des brennenden Magnesiums, um einen heißen, hellen Blitz zu erzeugen. Sie werden auch heute noch hergestellt [6] und in der analogen und digitalen Fotografie verwendet, da sie einzigartige Effekte erzeugen. Datenträger wie Papier, Magnetband oder analoger Film verlieren ihre Daten, wenn sie überhitzt oder sogar überbelichtet werden. Wenn ein Teil des Films um eine Blitzlampe gewickelt wird, kann er durch das Auslösen der Blitzlampe zerstört (verbrannt) werden. Wenn wir die Glühbirne in das Manipulationssichere Datenpaket einbauen, kann sie zur Warnung von Bob verwendet werden.

Alice schreibt also den speziellen Antwortcode auf ein Stück dünnes Papier (oder zeichnet ihn auf Magnetband auf), wickelt ihn um die Blitzbirne im Datenpaket, schaltet es scharf und schickt es an Bob. Wenn Eve während der Reise an der Box herumspielt, löst sie die Blitzlampe aus und zerstört damit den Code. Wenn Eve den Code nicht mehr lesen kann, ist sie nicht in der Lage, das Datenpaket wieder in seinen ursprünglichen Zustand zu versetzen. Und am Ende erfährt Bob von dem Manipulationsversuch.



Bild 1. Diese Magnesium-Blitzlichtbirne ist der Grundbaustein für unseren Manipulationsalarm.





Diese Schaltung ist nicht nur gegen Hardware-Trojaner wirksam, sondern wehrt auch Cold-Boot-Angriffe ab. Zwar wäre ein solcher Versuch wegen des Tieftemperaturschutzes des Datenpakets und der Tatsache, dass es das SRAM bei jedem Neustart löscht, sehr schwierig durchzuführen, aber ganz ausschließen kann man es nicht. Auch die Wiederherstellung des SRAM hilft nicht, wenn die Blitzbirne die auf einem physischen Träger enthaltenen Informationen zerstört.

#### Schaltungsbeschreibung

Der Schaltplan des blitzlichtgeschützten Analogspeicher-Addons ist in Bild 2 zu sehen. Verbinder K1 wird auf den Verbinder K3 der Hauptplatine [1] gesteckt. Mit Ausnahme der Stromversorgungsanschlüsse sind alle anderen Pins von K1 mit dem Mikrocontroller des Datenbox verbunden. Wenn das Datenpaket eine Manipulation feststellt und die MCU beschließt, ihren Speicher auf Null zu setzen, legt sie eine Binärkombination an die Eingänge von IC1, einem binären 4-zu-16-Standarddecoder, der den Ausgang *Fire* (Kathode von D3) aktiviert. Wurde das Datenpaket hingegen erfolgreich entriegelt, aktiviert die MCU stattdessen den Befehl *Block* (Kathode von D6). Nun kann das Datenpaket geöffnet, der Sicherheitsschalter SW3 geschlossen und die Blitzlampe aus K2 entfernt werden. Bob kann das Papier/Magnetband von der Blitzbirne abwickeln und den speziellen Antwortcode überprüfen. Die Ausgänge, die die Befehle *Fire* und *Block* aktivieren, werden von den Jumpern JP1 bis JP16 ausgewählt. Verwenden Sie nicht denselben Jumper für beide Befehle! Zusammen mit IC1 bilden sie einen Schutz gegen einen möglichen anwendungsspezifischen Trojaner, der versucht, den *Fire*-Befehl zu deaktivieren. Vor jedem Einsatz werden die Fire- und Block-Befehlscodes geändert, indem verschiedene Jumper auf der Platine gesetzt werden (**Bild 3**) und die Firmware der MCU entsprechend angepasst wird. Der anwendungsspezifische Trojaner von Mallory wird dann nicht funktionieren, weil er die Firmware oder die Ausgänge nicht mit Sicherheit erkennen kann. Ein Allzweck-Trojaner könnte die Codes vom SRAM in einen geheimen Flash-Speicher kopieren, aber nicht den Codes vom Papier oder Kassettenband, der um die Blitzbirne gewickelt ist.

IC2 ist als Komparator mit positiver Rückkopplung verdrahtet, um auf eine niedrige Spannung an seinem nicht-invertierenden Eingang zu reagieren. Die einstellbare Stromquelle IC4 liefert einen konstanten Strom für die Präzisions-Spannungsreferenz IC5, die eine 1,2-V-Referenzspannung für den Komparator IC2 liefert.

Wenn der *Fire*-Befehl aktiviert wird, wird der nicht-invertierende Eingang über die Diode D3 auf Low gezogen. Der Ausgang von IC2





# **Development Tools alle an einem Ort**

Tausende Tools von hunderten zuverlässigen Herstellern





Bild 4. Die Blitzlicht-Zusatzplatine ist auf die Unterseite der Hauptplatine gesteckt. Die beiden Drucktasten und der Schalter sind an der Seite zugänglich.

geht so auf Low, wodurch der Transistor T1 leitend wird. Die in C1 gespeicherte Ladung fließt durch L1 und D5, was zu einem Anstieg der Spannung an C5 führt, der höher als V<sub>d</sub> ist. Damit wird der Transistor T2 vollständig eingeschaltet. C5 hält T2 für mindestens 10...20 ms eingeschaltet, ausreichend Zeit, um die an K2 angeschlossene Blitzbirne zu zünden. C2 und C3 enthalten ausreichend Energie für einen 10-ms-Stromimpuls von 1 A bis 2 A, der benötigt wird, um das Magnesium im Inneren der Blitzlampe zu zünden.

Im Falle des *Block*-Befehls zieht die Diode D6 den invertierenden Eingang von IC2 auf 0,3 V, wodurch der Ausgang von IC2 auf high gesetzt wird. Der zerstörerische Stromkreis ist nun deaktiviert.

Mit dem Taster SW1 wird der *Fire*-Stromkreis durch Aufladen von C1 scharf geschaltet, Taster SW2 sorgt für eine Testzündung. SW3 ist ein Sicherheitsschalter, der die Zündung deaktiviert, indem er das Gate von T2 sicher auf 0 V hält.

LED1 und LED2 sind rote Warnleuchten. Wenn eine von ihnen oder beide leuchten, ist etwas nicht bereit, und Sie sollten keine Blitzlampe an K2 anschließen oder SW3 loslassen.

Im Gegensatz zur Datenbox selbst, die nur alle zwei Sekunden aus dem Stromsparmodus erwacht, wird diese Zusatzplatine ständig mit Strom versorgt, damit sie möglichst schnell auf Kurzschlüsse oder Spannungsabfälle durch Bohrattacken reagieren kann (siehe unten). Aus diesem Grund werden stromsparende ICs mit niedriger Betriebsspannung (bis zu 2,0 V) verwendet. IC1 zum Beispiel muss ein HC-Typ sein. So beläuft sich der zusätzliche Versorgungsstrom auf insgesamt nur 30  $\mu$ A. Die drei Kondensatoren (C1, C2 und C3) enthalten genug Energie, um die Blitzlampe auch ohne einen Befehl der MCU im Falle eines Kurzschlusses oder einer Unterspannung an V<sub>CC</sub> oder V<sub>bat</sub> auszulösen.

#### Anwendung

Stecken Sie die Zusatzplatine auf die Hauptplatine (**Bild 4**) und schalten Sie das gesamte System ein. Drücken Sie SW1, um den *Fire*-Stromkreis zu aktivieren. Wenn LED1 und LED2 erloschen sind, kann die Blitzbirne mit der eingewickelten geheimen Nachricht an K2 angeschlossen und mit SW3 freigegeben werden. Die gesamte Baugruppe wird dann in ein Gehäuse gelegt (**Bild 5**) und dieses kann nach dem in [1] beschriebenen Verfahren verschlossen werden.

#### **Bohrende Angriffe**

Was passiert nun, wenn Eve versucht, den Blitzlicht-Fireschaltkreis zu deaktivieren, indem sie das Gehäuse durchbohrt, zum Beispiel, um C1, C2 oder C3 mit einem Bohrer gegen Masse kurzzuschließen? Um dies zu verhindern, verfügt die Schaltung auch über einen barometrischen Luftdrucksensor (IC3, ein BME-280, der auch Temperatur und relative Luftfeuchtigkeit misst). Der Mikrocontroller liest alle zwei Sekunden den barometrischen (absoluten) Luftdruck von IC3 über den I<sup>2</sup>C-Bus (Pin 2 und Pin 3 von K1) ein.

Beim Scharfschalten der Box sollte ein Teil der Luft mit einer Vakuumpumpe abgepumpt werden. Der von IC3 gemessene Luftdruck wird von der MCU ausgelesen und temperaturkompensiert. Das Verhältnis zwischen Luftdruck und thermodynamischer Temperatur ist ausschlaggebend für die Auslösung der "Nullung" des Speichers. Da das Volumen im Gehäuse konstant ist, darf sich der Druck proportional zur thermodynamischen Temperatur im Bereich von -20°C bis +60°C (eigentlich 253 K bis 333 K) ändern. Dadurch wird ein Anbohren innerhalb von maximal zwei Sekunden erkannt (wenn die MCU aus dem Energiesparmodus erwacht).

Seien Sie vorsichtig, wenn Sie die Luft aus dem Gehäuse absaugen

- Bob soll ihn öffnen können, ohne dazu ein Brecheisen zu benötigen. Die Kraft, die Bob zum Öffnen des Deckels benötigt, lässt sich berechnen als  $F = A - \Delta p$ , wobei A die Fläche des Deckels und  $\Delta p$ die Druckdifferenz ist.

#### Kein Platz für Open-Source-Software

Der durch die Magnesium-Blitzbirne geschützte Speicher löst effektiv das Problem eines Allzweck-Hardwaretrojaners. Eve kann aus dem SRAM oder aus dem geheimen Flash-Speicher lesen, was sie will, aber das ist nutzlos, wenn der spezielle Antwortcode zusammen mit dem Papier oder dem Magnetband verbrannt ist. Ein anwendungsspezifischer Trojaner kann jedoch immer noch das Auslösen der Blitzlampe verhindern, so dass auch diese Bedrohung abgeschwächt werden muss.

Bis hierher haben wir die Verwendung von Open-Source-Software empfohlen, wann immer dies möglich ist. Obwohl Open-Source-Software aus vielen Gründen gut ist, sind wir hier auf einen Fall gestoßen, in dem Open-Source-Software nicht verwendet werden sollte. Ein anwendungsspezifischer Trojaner ist viel komplizierter als ein Allzweck-Trojaner. Er kann nur dann funktionieren, wenn er die Firmware mit Sicherheit erkennen kann. Eine Möglichkeit für Alice, dieser Bedrohung zu begegnen, besteht darin, die Firmware selbst zu schreiben. Sie wird dies auf ihrem speziellen Computer tun, der nie mit dem Internet verbunden war, und sie wird denselben Computer zur Programmierung der MCU verwenden. Wenn Mallory nichts über die Firmware weiß, kann ihr Trojaner sie auch nicht erkennen und manipulieren.

#### Weitere Verbesserungen

Abgesehen davon, dass wir auf quelloffene Software verzichten, können wir auch eine kleine Einschränkung unserer Open-Hardware-Prinzipien

#### **Dramatis Personae**

In der Kryptographieliteratur findet die Kommunikation normalerweise zwischen Alice (A) und Bob (B) statt. Die abhörende Eve versucht, passiv mitzuhören, ohne sich einzumischen, während die böswillige Mallory nicht zögert, Nachrichten zu verändern, zu ersetzen oder wiederzugeben.

#### Sicherheitsvorkehrungen

Mg-Blitzbirnen arbeiten mit sehr niedrigen Spannungen - schon 1,0 V können ausreichen, um sie zu entzünden. Das Magnesium im Inneren der Blitzbirne verbrennt bei Temperaturen von über 3000 °C, so dass der Glaskolben aufgrund dieser extremen thermischen Belastung bricht und manchmal zersplittert. Die Splitter dieses Glases sind nicht scharf, es besteht also keine Verletzungsgefahr. Aus diesem Grund kann die Kunststoffbeschichtung des Glaskolbens, falls vorhanden, entfernt werden, um die Wärmeübertragung zu verbessern und die Zerstörung des ihn umgebenden Mediums zu gewährleisten. Zwar können geschmolzene Metallstücke bei diesen extremen Temperaturen Verbrennungen verursachen, andererseits ist die im Manipulationssicheren Datenpaket gezündete Blitzbirne vollkommen sicher - viele herkömmliche elektronische Geräte setzen viel mehr Energie frei, wenn sie durchbrennen und während des Betriebs ausfallen (wenn auch selten bei so hohen Temperaturen). Eine Fehlzündung einer Mg-Blitzbirne in der Nähe Ihrer ungeschützten Augen kann allerdings sehr unangenehm sein und sogar zur vorübergehenden Erblindung führen (so funktionieren Blendgranaten). Wickeln Sie deshalb Papier oder Klebeband um die Blitzbirne, **bevor** Sie sie an den K2 anschließen, und reduzieren so die Helligkeit.

#### Papier-Tipps

Wenn Papier als Medium zum Schreiben des Geheimcodes verwendet wird, dann empfiehlt sich dünnes Zigarettenpapier. Es empfiehlt sich auch, es in eine von Kaliumpermanganat gesättigte Wasserlösung zu tauchen. Ein solches starkes Oxidationsmittel



Bild 5. Das Manipulationssichere Datenpaket mit dem Blitzlichtschutz ist bereit, an Bob geschickt zu werden.



# STÜCKLISTE

#### Widerstände:

 $\begin{array}{l} (alle 5\%, 0805, 0,125 \ W) \\ R1,R2,R3 = 100 \ k \\ R4,R15 = 10 \ k \\ R5,R8, R11 = 220 \ \Omega \\ R6,R14 = 1 \ M \\ R7,R10,R12 = 1 \ k \\ R9 = 330 \ k \\ R13 = 560 \ k \\ R16 = 5k1 (5k6 \ || \ 56 \ k) \end{array}$ 

#### Kondensatoren:

C1 = 10 µ (1206) C2,C3 = 2200 µ, 6V3, Raster 5 mm C4,C6,C7 = 100 n (0805) C5 = 47 n (0805)

#### Induktivität:

 $L1 = 1 \text{ mH}, 10 \Omega$ 



D1,D3,D4 = 1N4148W D2,D5,D6 = BAT54 IC1 = 74HC154D IC2 = MAX9915 IC3 = BME280 IC4 = LM234Z-3 IC5 = LM285D-1.2 LED1,LED2 = LED, rot, 0805 T1 = BC857 T2 = BSS205N

#### Außerdem:

K1 = 1x10-polige Stiftleiste, Raster 0,1" K2 = 2-polige Platinen-Anschlussklemme, Raster 0,2", Wago 236-402 SW1,SW2 = Drucktaster, RS282G05A3 SW3 = Schiebeschalter, SS12SDP2





#### WEBLINKS

[1] Luka Matić: "Manipulationssicheres Datenpaket", ElektorMag Mai/Juni 2020: www.elektormagazine.de/180445

- [2] Einschleusen eines Hardware-Trojaners auf einen Siliziumchip:
- www.schneier.com/blog/archives/2018/03/adding\_backdoor.html
- [3] Datenremanenz im SRAM bei niedrigen Temperaturen: www.cl.cam.ac.uk/techreports/UCAM-CL-TR-536.pdf
- [4] Löschen von Daten aus Flash-Laufwerken: www.schneier.com/blog/archives/2011/03/erasing\_data\_fr.html
- [5] Datenremanenz im RAM-Speicher: www.usenix.org/legacy/events/sec01/full\_papers/gutmann/gutmann.pdf
- [6] Meggaflash, Hersteller von Magnesium-Blitzlampen: www.meggaflash.com/

[7] Dieses Projekt auf Elektor Labs:

- www.elektormagazine.de/labs/magnesium-bulb-analogue-memory-add-on-to-the-tamper-evident-box
- [8] Manipulationssichere Box auf Elektor Labs:
- www.elektormagazine.de/labs/tamper-evident-paper-mail-box-for-secure-distribution-of-one-time-pads [9] Interview mit Luka Matić im Elektor TV:
- www.elektormagazine.com/news/secure-communications-an-interview-with-luka-matic

dulden. Alice kann die Sicherheit verbessern, indem sie auch die Verbindungen zwischen K1 und IC1 verändert, zum Beispiel durch Lötbrücken wie bei JP1...JP16. Die Herstellung einer selbst entworfenen Platine kann, da Eve und Mallory eifrige Elektor-Leser sind, die Robustheit gegen Mallorys Hardware-Trojaner und Eves Bohrangriffe nur verbessern.

Auch wenn das Blitzbirnen-Add-on gegen Kurzschlüsse mittels Bohrangriffs geschützt ist, sollte man bedenken, dass Eve ein theoretisches Zeitfenster von bis zu zwei Sekunden (bis die MCU aufwacht und eine Änderung des Luftdrucks feststellt) hat, um einen solchen Angriff durchzuführen. Das bedeutet, dass anfällige Leiterbahnen (vor allem V<sub>d</sub>) mechanisch gegen Bohrangriffe verstärkt werden sollten, etwa mit dünnen Stäben einer sehr harten Legierung, die in einen Körper aus weichem Metall (Messing oder Aluminium) eingebettet werden, um einen Bohrer zu brechen oder ihn abzulenken - ähnlich wie bei Hochsicherheitsschlössern.

#### Verbesserte Sicherheit

Der in diesem Artikel beschriebene einfache und billige Zusatz verbessert die Sicherheit der Originalschaltung erheblich und verhindert viele High-Tech-Angriffe. Jede Art von Magnetband, die mit diesem System getestet wurde, wurde von einer AG-1-Magnesiumblitzlampe verbrannt. Auch alle Tests mit Papier waren erfolgreich.

Für unsere Freunde Alice und Bob, die mit einem schmalen Budget auskommen müssen, bleibt alles billig und selbstgemacht. Preiswertes analoges Low-Tech kann teures digitales High-Tech schlagen, wenn

es sorgfältig eingesetzt und kombiniert wird. Es mag kontraintuitiv erscheinen, aber der Intuition des "gesunden Menschenverstands" ist erstaunlich oft nicht zu trauen. Deshalb müssen, wie bei jedem "sicheren" Gerät, neue Angriffs- und Verteidigungsmöglichkeiten von mehreren unabhängigen Experten erforscht werden, um die Gesamtsicherheit umfassend zu bewerten.

Alle Designdateien für dieses Projekt und vieles mehr finden Sie unter [7], [8] und [1]. Ein Blick auf [9] kann helfen, Ihre paranoiden Fähigkeiten zu entwickeln. Auf dass Eve und Mallory sich (an der Blitzbirne) die Finger verbrennen!

200647-03

#### **Ein Beitrag von**

Idee, Gestaltung und Text: Luka Matić Platinenentwurf und Redaktion: Clemens Valens Übersetzung: Rolf Gerstendorf Layout: Harmen Heida

#### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an die Redaktion unter clemens.valens@elektor.com.



MEDSER **Medical Services** 

## LUST, DIESEN CT **ZU REPARIEREN?**

Dann bewerben Sie sich als

#### ELEKTRONIKTECHNIKER/-INGENIEUR (m/w/d)

Sie finden und beheben Defekte mit Herzblut, elektronische Schaltungs- und Messtechnik sind für Sie wie erweiterte Körperteile und Verantwortungsbewusstsein sowie wissenschaftliches Vorgehen sind Teil Ihrer DNA? Dann sollten wir uns unterhalten, denn wir suchen Menschen wie Sie.

Ihre Fragen beantwortet unsere Bewerberhotline gerne schon vorab: +49 61 04 408 84-212 oder karriere@medser.de. Auch nach Feierabend.





# Elektor 2MHz LCR-Meter Kit

> 1 GΩ

 $\Phi = 10.577^{\circ}$  Q = 0.00000

R-Hold OFF AC 1V RMS

#### Alle notwendigen Bauteile für den Zusammenbau und die Kalibrierung sind enthalten:

- 2 bestückte Platinen: Mainboard + Display (Drucktasten, Drehimpulsgeber und LCD)
- > 4 BNC-Buchsen (nicht bestückt), 4 Jumper

elektormagazine.de/pesters

- Kelvin-Kabel mit Clips und 4 Steckern
- > Mini-USB/USB-A-Kabel
- > 24-poliges Flachbandkabel (15 cm)
- > Aluminiumknopf für Drehimpulsgeber
- > Einstellwerkzeug für Trimmer

#### Mehr Infos & Bestellung:

www.elektor.de/19883



Besonderes Augenmerk wird auf die einfache Kalibrierung und Bedienung gelegt.

PARALL

TRIM

AUG F

R-HOLD

MENU

20 1

14.8 H 100 Hz

U× = 1.02 V I× = 20.5 μA

Average SLOW DC 0.0 V

ektor 2 MHz LCR Meter

Hammond-Gehäuse gebohrt und gefräst, mit bedruckten Front- und Seitenwänden



# **Bluetooth-Beacons** in der Praxis

### Ortsbestimmung in Innenräumen

#### Von Dr. Veikko Krypczyk

Ortung außerhalb von Gebäuden basiert üblicherweise auf GPS und/oder der Mobilfunktechnologie. Doch wie ortet man Personen und Gegenstände innerhalb von Gebäuden, wo keine Satellitensignale eindringen und keine Mobilfunkverbindung gegeben ist? Dort kann man so genannte Beacons auf Basis der Bluetooth Low Energy-Technik einsetzen. Schauen wir uns an, wie das funktioniert!

Cuxhavener Kugelbake von 1703 (© A.Savin, WikiCommons, https://de.wikipedia.org/ wiki/Kugelbake#/media/Datei:Cuxhaven\_07-2016\_ photo23\_Kugelbake.jpg)



Als Bake (englisch: beacon) wurde vor Jahrhunderten eine natürliche oder künstlich errichtete optische Landmarke bezeichnet, die den Seeleuten den rechten Weg in den gefährlichen Küstengewässern wies. Mit dem Aufkommen den Funkverkehrs in der Seefahrt Ende des 19. Jahrhunderts wurden solche stationären, meist in Ketten angeordnete Baken mit Sendern ausgestattet (Seefunkfeuer). Die einzelnen Baken sendeten in Intervallen ungerichtete individuelle Morsecodes zur Identifikation, so dass sich ein Schiff an der Bakenkette "entlanghangeln" konnte. Was den Seefahrern recht war, sollte den Fliegern nur billig sein: Aus den ersten Versuchen in den 1920er Jahren, mit Hilfe des Funks den Landeanflug bei widrigen Wetterbedingungen abzusichern, entwickelte der böhmische Ingenieur Ernst Ludwig Kramar ein Instrumentenlandesystem im unteren UKW-Bereich, die so genannte Lorenz-Bake [1], die Anfang der 30er Jahre weltweit auf Flughäfen installiert wurde. Während in der Seefahrt Funkbaken mit dem Aufkommen der GPS-Navigation ihre Bedeutung verloren (ausgenommen Notrufbaken), stehen Funkbaken der unterschiedlichsten Ausprägungen in der Luftfahrt immer noch hoch im Kurs.

Zu Beginn des 21. Jahrhunderts, im Handy-Zeitalter, kam eine neue Form von Beacons hinzu, die den ursprünglichen Seefunkfeuern ähnelt: Mehrere Beacons senden individuelle Kennungen, die von einem mobilen Gerät empfangen werden. Anhand der Signalstärke der eintreffenden Kennungen kann eine geeignete Software bestimmen, an welchem Ort relativ zu den empfangenen Beacons man (das Smartphone, Tablet oder Wearable) sich befindet. Natürlich wird dazu kein UKW-Funk verwendet, sondern das energiesparende Übertragungsprinzip von Bluetooth Low Energy (BLE). Kommt Ihnen alles bekannt vor? Tatsächlich, das Verfahren ist brandaktuell, denn genau darauf basiert unsere Corona-Warn-App!

In diesem Artikel wollen wir in die Technologie eindringen, die aus Hardware (Beacons) und Software (Konfiguration und Anwendung) besteht. In ersten Experimenten machen wir uns mit der Thematik vertraut und schaffen damit die Basis für eigene Anwendungen.

#### Vielfältige Anwendungsoptionen

Abstandsmessungen und Ortung innerhalb von Gebäuden sind die typischen Einsatzgebiete von Beacons. Durch Kombination von Beacons sind die unterschiedlichsten Einsatzszenarien (Use Cases) im privaten Alltag als auch im Business, im geobasierten Marketing, mobilen Shopping und diversen Anwendungen in den Bereichen Mixed- und Augmented-Reality denkbar. Dabei können beispielsweise relevante Informationen oder Werbung an einem bestimmten Ort innerhalb eines Gebäudes übermittelt werden, wenn der Nutzer sich gerade in der Nähe (Region) aufhält.

> Mobile Shopping: Auf dem Smartphone eines Kunden lassen sich Produkt-



Bild 1. Use Case der Beacon-Nutzung.

informationen, aktuelle Angebote, Rabatt-Codes und so weiter zu einzelnen Artikeln einblenden. Ebenso ist es denkbar, dem Nutzer weitere Produkte anzubieten, wenn er sich für eine bestimmte Produktkategorie interessiert. Die Kunden können auf diese Weise durch das Produktangebot geleitet werden. So haben haben die Einzelhändler die Möglichkeit, über die Beacons mit dem Smartphone der Kunden in Kontakt zu treten. Die Marktforschung profitiert von den fortschreitenden Positionsdaten der Kunden, um die Produktanordnung in großen Märkten zu optimieren. Dabei kann man sich das in Bild 1 dargestellte Szenario vorstellen. Das Beacon sendet ein Signal über BLE (Schritt 1). Dieses wird vom Smartphone empfangen (Schritt 2). Daraufhin wird die passende App aus dem Hintergrund geweckt (Schritt 3). Die App registriert und ordnet das Beacon anhand seiner ID zu (Schritt 4), fordert weitere Informationen über das Backend an (Schritt 5) und präsentiert diese auf dem Screen des Smartphones (Schritt 6).

> Navigation: Beacons werden bei der Navigation in Gebäuden, beispielsweise in Museen, Bahnhöfen, Flughäfen oder Sportstadien eingesetzt. Nach der Platzierung an verschiedenen Stellen im Gebäude werden per Bluetooth Signale an die mobilen Endgeräte gesendet. So kann die Position der mobilen Geräte

und ihrer Nutzer kontinuierlich geortet und an das Indoor-Navigationssystem weitergegeben werden. Für die Interpretation der Signale wird eine App auf dem Smartphone benötigt. Die Anzahl der Beacons im Gebäude hängt von der Fläche und von der gewünschten Genauigkeit ab.

- > Unterstützung von Barrierefreiheit: Durch eine gezielte Sprachausgabe und Bildanweisungen, beispielsweise auf einem Monitor, kann man Menschen mit Behinderungen an komplexen oder stark frequentierten Plätzen helfen, sich besser zurechtzufinden.
- > *Logistik*: Durch die Platzierung von Beacons entlang der Versorgungs- oder Logistikkette besteht die Möglichkeit, Beacons zusammen mit anderen modernen Techniken wie RFID oder Real Time Fleet Management sinnvoll zu kombinieren und damit die Durchführung der Logistikaufgaben zu optimieren.

Ein besonderer Vorteil der Beacon-Technologie ist der geringe Energiebedarf, der einen lang andauernden Betrieb ohne externe Stromversorgung möglich macht. Kritisch ist der Einsatz von Beacons dagegen aus datenschutzrechtlichen Gründen, da darüber das Verhalten (Position, Laufwege, Aufenthaltszeiten) des Nutzers analysiert werden kann. Andererseits müssen die Nutzer einer Anwendung aktiv zustimmen, denn für die Ortung wird eine eigene Software auf dem mobilen Endgerät benötigt. Die Beacons



#### **BEACONS UND DATENSCHUTZ**

Beacons senden zunächst nur **Informationen ohne Personenbezug** und können selbst keine Daten empfangen oder verarbeiten. Insofern bestehen keine datenschutzrechtlichen Bedenken beim Senden von Signalen. Ein Personenbezug wäre allenfalls dann denkbar, wenn ein Beacon einer bestimmten Person zugeordnet ist (die zum Beispiel ein Beacon zur Identifikation mit sich trägt), sofern weitere Informationen des Nutzers mit diesem Datum verbunden werden.

Im Gegensatz zum Beacon selbst werden Informationen durch die installierte App auf dem Smartphone empfangen und verarbeitet. Da die betreffende App in den meisten Fällen mit einem Nutzerkonto verbunden sein soll, besteht hier durchaus ein Personenbezug. So kann zum Beispiel das Programm die Information auswerten, wo sich ein bestimmter Kunde wie lange an welchem Regal aufgehalten hat oder auch das Kaufverhalten analysiert werden. Durch diesen Personenbezug ist eine Verarbeitung der Daten grundsätzlich nur mit einer Einwilligung des Betroffenen möglich. Da der Betroffene die notwendige App aktiv auf seinem Smartphone installieren muss, sollte dieser zu Beginn des Nutzungsvorgangs in einer Datenschutzerklärung datenschutzkonform informiert werden. Insbesondere muss er in diesem Zusammenhang über Zweck und Marketingmaßnahmen informiert werden (Art. 13 DSGVO).

Die datenschutzrechtliche Verantwortung liegt also grundsätzlich bei den Anbietern der Apps, die Informationen von Beacons empfangen und diese verarbeiten. Informiert der Anbieter seine Nutzer ausreichend und holt seine Einwilligung datenschutzkonform ein, sollte dem Einsatz von Beacon nichts im Wege stehen. Nach David Oberbeck, Datenschutzkanzlei [19].

selber senden lediglich die Signale, es sind üblicherweise keine Empfänger (siehe auch **Kasten** "Beacons und Datenschutz").

#### **Bluetooth Low Energy als Basis**

Die BLE-Technologie stammt aus dem Jahr 2006 und wurde von Nokia zur unidirektionalen Übertragung von Daten eingeführt. BLE-Geräte benötigen weitaus weniger Energie als herkömmliche Bluetooth-Geräte. Im Durchschnitt hält die Batterie eines Beacons mehrere Jahre. Auch die Herstellung der BLE-Module ist im Vergleich zu klassischen Bluetooth-Devices deutlich günstiger. Im Jahr 2013 machte Apple die Beacon-Technologie der breiten Öffentlichkeit zugänglich. Hersteller und Entwickler, welche die Beacon-Technologie verwenden wollten, mussten zunächst eine Lizenz bei Apple beantragen. Anschließend konnten das *Software Development Kit* (SDK) und die Programmbibliotheken genutzt werden. Auf dem Markt gibt es heutzutage zahlreiche Unternehmen, die sich auf die Produktion von Beacon-Signalgebern spezialisiert haben. Es sind drei Beacon-Typen mit unterschiedlichen Datenformaten verfügbar, welche sich bezüglich des Datenformates unterscheiden:

 iBeacon: Es handelt sich um das von Apple definierte Format. Unter [2] findet man die Einführung in das Thema und

### Tabelle 1. Major und Minor dienen der Zuordnungvon Region und Subregion zu einem Beacon [14].

Store Lo	ocation	Größe	San Francisco	Paris	London
UUID		16 Bytes	D9B9EC1F-3925	5-43D0-80A9-1E	39D4CEA95C
Major		2 Bytes	1	2	3
	Bekleidung		10	10	10
Minor	Haushalt	2 Bytes	20	20	20
	Elektro		30	30	30

die vollständige Spezifikation.

- > AltBeacon: Es handelt sich um ein offenes Format, das ohne Lizenzgebühren implementiert werden darf. Die Dokumentation findet man unter [3].
- Eddystone: Dies ist ein von Google definiertes, offenes Format. Mehr Informationen dazu findet man unter [4].

Wie funktionieren Beacons? Die BLE-fähigen Beacons senden in dem stromsparenden Bluetooth-Modus kurze Nachrichten in einem festen Intervall, die von Bluetooth-fähigen Geräten in der unmittelbaren Umgebung empfangen werden können. Dabei handelt es sich um ein so genanntes Broadcasting-Verfahren. Ebenso kann jedes Smartphone, jedes Tablet oder jede Smartwatch mit BLE-Einheit als potenzieller Beacon angesehen werden. Einen wichtigen Unterschied gibt es jedoch: Ein Smartphone oder ein Tablet kann BLE-Signale senden und Bluetooth empfangen. Beacons dagegen senden ausschließlich die Signale in kurzen Zeitintervallen, so dass eine Signalregion entsteht. Damit man die entstehenden Daten nutzen kann, muss auf dem Smartphone eine entsprechende Software (App) installiert sein.

#### Spezifikationen

Welche Informationen werden durch ein Beacon übermittelt? Hier müssen wir zwischen den Beacon-Typen *iBeacon, AltBeacon* und *Eddystone* unterscheiden (**Bild 2**). Beginnen wir mit der Datenspezifikation der *iBeacons* (Bild 2a). Dabei bedeuten die einzelnen Werte [5]:

- Prefix: Es enthält die Hex-Daten: 0x 020106 1AFF 004C 02 15. Dabei definiert 0x 020106 die generelle Information, dass das sendende Gerät nicht mit High Speed kompatibel ist und nur ein Broadcasting (Senden) und keine Verbindung (bidirektional) zulässt. 0x1AFF besagt, dass die folgenden Daten 26 Byte lang und herstellerspezifisch sind. 0x004C ist Apples Bluetooth-Signal-ID. 0x02 ist eine sekundäre ID und 0x15 definiert die verbleibende Länge als 21 Bytes (16+2+2+1).
- > UUID (Universally Unique Identifier): Der Wert ist 16 Byte lang und wird verwendet, um für eine eindeutige Zuordnung der gesendeten Daten zum iBeacon in Systemen mit mehreren Komponenten zu sorgen.
- Major: Der Major-Wert ist 2 Byte groß.
   Dieser Wert hat die Funktion, die Signalregion zu definieren.

- > Minor: Der Minor-Wert ist ebenso 2 Byte groß. Dieser Wert bestimmt eine Subregion.
- > TX-Power: Dieser Wert gibt die potentielle Signalstärke an. Er wird von der App dazu verwendet, um zu bestimmen, wie weit das Beacon vom Smartphone entfernt ist. Bitte beachten Sie, dass der Wert TxPower vom Benutzer kalibriert werden muss, um genau zu sein.

Ein Beispiel für eine Zuordnung der Daten von Major und Minor für eine Indoor-Navigation in einem Kaufhaus zeigt Tabelle 1. Kommen wir zur Spezifikation des Datenformates für die AltBeacon-Typen (Bild 2b). Die AltBeacon-Spezifikation ist 28 Byte lang, wobei 26 Byte durch den Nutzer anpassbar sind. Die ersten beiden Bytes werden vom BLE-Stack festgelegt. ADV Length und ADV Typ geben die Länge des Datenpaketes und den Typ an. Alle anderen Werte können konfiguriert werden.

Bei den Beacons vom Typ Eddystone schließlich können verschiedene Varianten definiert werden (Bild 2c), die kombiniert oder einzeln verwendet werden können. Es bedeuten:

- > UUID: Es wird eine eindeutige 16-Byte-Beacon-ID übermittelt.
- > EID: Dieser Frame sendet eine Kennung, die nur kurzlebig und dazu verschlüsselt ist, um die Datensicherheit zu erhöhen.
- > URL: Es wird eine URL übermittelt. Nutzer können die URLs verwenden, um auf Inhalte im Internet zuzugreifen.
- > TLM: Reserviert für die Übermittlung von Telemetrie-Status-Daten über das Beacon-Gerät selbst. Diese Informationen umfassen die Batteriespannung, die Anzahl der Broadcast-Pakete und die Gerätetemperatur.

Entwickelt man Applikationen mit Beacons, muss man den Beacon-Typ berücksichtigen oder durch die Software für eine Auswahloption sorgen.

#### Ortung

Die App auf dem mobilen Endgerät identifiziert die Signalgeber und berechnet die Abstände zwischen den Sendern und dem Empfänger anhand der ankommenden Sendeleistung. Die Sendeleistung wirkt sich direkt auf die Signalreichweite aus. Je mehr Leistung, desto größer die Reichweite. Der Received Signal Strength Indicator (RSSI) bezeichnet die Stärke des Beacon-Signals beim Smartphone. Die Entfernung wird



Bild 2. Spezifikationen der Beacon-Typen ([15], [16], [17], [18]).



Version 1 Byte

Type 1 Byte





Time 4Byte:





Bild 4. Beacon EMBC01 der Firma EM Microelectronic.

bestimmt, indem die empfangene Signalstärke ins Verhältnis zum (laut Datenblatt) erwarteten RSSI-Wert in einem Meter Abstand gesetzt wird. Die Skalierung der RSSI-Werte wird dabei den Chipherstellern überlassen: Aus den Datenblättern kann entnommen werden, wie der RSSI-Wert in Leistung umgerechnet werden kann. Die Angabe erfolgt in der Einheit Dezibel Milliwatt (dBm). Folgende Formel beschreibt üblicherweise den Zusammenhang zwischen dem RSSI-Wert und der Entfernung [6], [7]:  $RSSI = -(10 n \log 10 d + a)$ 

mit n: umgebungsabhängiger Dämpfungswert a: empfangene Signalstärke mit Sender in einem Meter Entfernung (Referenzsignalstärke laut Datenblatt) und

Bild 5. Anzeige der Entfernung in der App EMBC-Finder.

d: Distanz zwischen Sender und Empfänger. Für eine Indoor-Umgebung wird ein Wert von n = 3 angenommen. Man muss dabei beachten, dass die Genauigkeit der Messungen und damit auch die Genauigkeit der Distanzbestimmung umso schlechter ist, je weiter Sender und Empfänger von einander entfernt sind.

Um die Ortsbestimmung exakt durchführen zu können, sind mindestens drei Beacons notwendig (**Bild 3**). Ähnlich wie bei GPS werden die Abstände zwischen den Einheiten nach dem *Laterations*-Prinzip gemessen. Die Datenübertragung bei Beacons ist jedoch auf den Nahbereich ausgelegt. Es wird zwischen *Immidiate* (weniger als 50 cm), *Near* (bis 3 m), *Far* (mehr als 10 m) und *Unknown* unterschieden. Leistungsstarke Beacons können Signale bis zu 450 m weit senden. Durch Hindernisse



Bild 6. Die Komponenten TBeacon und TBeaconDevice kapseln die gesamte Komplexität.



Bild 7. Konfiguration der Komponente TBeacon.

kann sich die Reichweite allerdings verringern. Flexible Regionen (Geofences), die durch die Positionierung der Beacons immer wieder verändert werden können, werden anhand von Latitude, Longitude und einem Radius definiert.

#### **Ein erstes Experiment**

Nach diesen notwendigen theoretischen Betrachtungen wollen wir uns ansehen, wie Beacons in eigenen Projekten arbeiten können. Zunächst benötigen wir einen oder mehrere Beacons (Bild 4), die man in vielen Variationen von [8] (europaweit) beziehen kann.

Das Modell EMBC01 ist ein Bluetooth Low-Energy-Beacon, das speziell zur Distanzmessung entwickelt wurde. Die übermittelten Daten sind nach dem Apple-Standard UUID, Major ID und Minor ID. Der EMBC01 ist in einem wetterfesten Gehäuse mit einer (auswechselbaren) Knopfzellenbatterie verbaut. Mit einem Druckknopf kann man zwischen den unterschiedlichen Modi wechseln:

- > Sleep Mode: Lagermodus mit einer typischen Haltbarkeit der Batterie von über sieben Jahren.
- > ID Short Range Mode: Mit einem Sendeintervall von 100 ms, einer Reichweite von 15 m und einer typischen Batterielebensdauer von 1,5 Monaten.
- > ID Medium Range Mode: Mit einem 500-ms-Intervall, 30 m Reichweite und einer typischen Batterielebensdauer von 7,5 Monaten.
- > ID Long Range Mode: Mit einem Intervall von 1 s, einer typischen Reichweite von 75 m und einer typischen Batterielebensdauer von 12,5 Monaten.

Auf dem Gehäuse des EMBC01 sind eine Seriennummer und ein QR-Code aufgedruckt. Dieser Typ von Beacons ist mit einer eindeutig definierten UUID, Major ID und Minor ID vorprogrammiert und kann zur Sicherheit nicht drahtlos neu programmiert werden. Um die Funktion zu demonstrieren, benötigt man eine App auf dem Smartphone, unter Android zum Beispiel die App EMBC-Finder und unter iOS die App Locate. Wählen Sie in der App einen aktiv sendenden Beacon aus, dann können Sie seine Entfernung ermitteln (Bild 5).

#### **Entwicklung eigener Software**

Um eigene Anwendungen der Beacon-Technologie umzusetzen, muss man auch ein eigenes Programm für das mobile BLE-Gerät, kurz gesagt eine App, entwickeln. Dazu kann man beispielsweise die kostenfreie Community Edition von RAD Studio verwenden [9]. Mit Hilfe der integrierten Entwicklungsumgebung lassen sich unter anderem Apps für iOS und Android, aber auch Applikationen für Desktop-Systeme (Windows, macOS, Linux) aus einer gemeinsamen Quellcodebasis erstellen [10]. Dank spezieller Komponenten für die Beacon-Technologie muss sich der Entwickler keine Gedanken über die Auswertung der BLE-Signale und der Protokolle machen. Die Komponenten werden in der Entwicklungsumgebung konfiguriert und stehen dann unmittelbar im Programmcode zur Verfügung. RAD Studio bietet für die Arbeit mit der Beacon-Technologie die beiden folgenden Komponenten (Bild 6) [11]:

- > TBeacon: überwacht eine Liste der angegebenen Beacon-Regionen und verwaltet die Informationen über die zugehörigen Ereignisse der Beacons; unter anderem werden Änderungen in der Entfernung erkannt.
- > TBeaconDevice: kann ein BLE-Gerät als reguläres Beacon verwenden. Sie können eine TBeaconDevice-Komponente in einer Anwendung konfigurieren, die auf einem BLE-Gerät ausgeführt wird, so dass dieses Gerät Proximity-Daten sendet. Die Werte für die Eigenschaften UUID, Major, Minor und TxPower können konfiguriert werden.

Um externe Sender (Beacons) zu nutzen, verwendet man die TBeacon-Komponente. Mit der Komponente TBeaconDevice lässt sich ein Smartphone in ein aktives Beacon verwandeln. Mit wenigen Schritten kann man die Funktionsweise demonstrieren: Auf GitHub findet man unter [12] geeignete Quellcodebeispiele. Klonen Sie dazu das komplette Repository auf Ihren Rechner und wechseln Sie in das Unterverzeichnis: RADStudio10.3. 3Demos/Object Pascal/Multi-Device Samples/Device Sensors and Services/ Bluetooth/Beacons/ExtendedBeaconScanner/. Öffnen Sie das Projekt in RAD Studio.

In Bild 7 lassen sich Konfiguration und Vorgehen bei der Programmierung sofort erkennen. Die Komponente TBeacon wird hier so konfiguriert, dass sie alle Arten von Beacons (iBeacon, AltBeacon und Eddystone) erkennt. Der Scanvorgang wird über die Prozedur StartScan aufgerufen. Untersuchen Sie dazu den Quellcode. Sofern sich sendende



Bild 8. Die ermittelten Beacons werden angezeigt.

Beacons in der näheren Umgebung befinden, werden diese auf der Oberfläche nach dem Start des Programms angezeigt (Bild 8). Für eine eigene Anwendung können die Daten entsprechend auf dem Smartphone ausgewertet werden.

#### **Komplexe Ortungsszenarien**

Für eine simple Aufgabe, etwa um zu ermitteln, ob sich ein mobiles Gerät in der Nähe eines Beacons befindet, bedarf es keiner umfassenden Planung. Komplizierter wird es aber, wenn man beispielsweise in einem Ladengeschäft den Raum in Zonen (Beacon-Fence) einteilen möchte. In diesem Fall müssen zur Überwachung der Zonen die Beacons entsprechend platziert werden. Nutzer können dann anhand dieser geographischen Anordnung geortet und das Betreten oder Verlassen einer Zone kann festgestellt werden. Dazu kann man einen grafischen Editor verwenden, zum Beispiel den BeaconFence-Karteneditor von RAD Studio [13]. Über einen solchen Karteneditor kann man das Layout der Räume und die Anordnung der Beacons, Zonen und Pfade definieren. Es können Aktionen dafür festgelegt werden, wenn ein BLE-fähiges Endgerät in eine Zone eintritt, die durch ein oder mehrere Beacons aufgespannt wird. Dieses grafische Vorgehen erleichtert die Umsetzung von





Business Cases erheblich, denn sonst müsste man die gesamte Anordnung und Konfiguration der Beacons manuell durchführen.

#### **Fazit und Ausblick**

Beacons basieren auf der BLE-Technik und bieten die Möglichkeit, die Position von Kunden, Nutzern, Besuchern oder bewegten Gegenständen in einem Raum zu bestimmen. Die Protokolle sind standardisiert. Wir können auf fertige Komponenten zur Auswertung der empfangenen Daten zurückgreifen, so dass sich der Programmieraufwand in Grenzen hält. Am Anfang eines solchen Projektes steht immer eine Idee (*Was soll Ziel der Indoor-Ortung sein?*), dann die Konzeption (*Wo und wie werden die Beacons angeordnet?*) und abschließend kann man sich schrittweise über die softwaretechnische Umsetzung Gedanken machen. Dabei unterstützen fertige Bibliotheken den Softwareentwickler. Interessante Szenarien im "Dreiländereck" zwischen Spiel, Spaß und ernsthaften Anwendungen warten darauf, realisiert zu werden.

200550-01

#### Sie haben Fragen oder Kommentare?

Haben Sie Fragen oder Anmerkungen zu diesem Artikel? Senden Sie eine E-Mail an den Autor über die Redaktion von Elektor: redaktion@elektor.com.

#### Ein Beitrag von

Text und Grafik: Dr. Veikko Krypczyk

Redaktion: Rolf Gerstendorf

Layout: Giel Dols

#### WEBLINKS

- [1] Lorenz-Bake: https://de.wikipedia.org/wiki/Lorenzbake
- [2] Beacon-Datenformat: https://developer.apple.com/ibeacon/
- [3] Quelloffenes AltBeacon-Datenformat: https://altbeacon.org/
- [4] Google-Beacons: https://github.com/google/eddystone/blob/master/protocol-specification.md
- [5] Beacons im Vergleich: https://austinblackstoneengineering.com/ble-beacons-ibeacon-altbeacon-uribeacon-and-derivatives/
- [6] Anton Anders: Indoor-Positionsbestimmung mit Hilfe von Bluetooth-Low-Energy-Beacons und Pedestrian Dead Reckoning: https://cse.cs.ovgu.de/cse-wordpress/wp-content/uploads/2016/08/BA\_Anton\_Anders.pdf
- [7] Sendeleistung und Distanz: https://community.estimote.com/hc/en-us/articles/201636913-What-are-Broadcasting-Power-RS-SI-and-other-characteristics-of-beacon-s-signal-
- [8] Beacons für Europa : www.beaconshop24.de/
- [9] RAD Studio: www.embarcadero.com/products/rad-studio
- [10] Krypczyk, Veikko: Mobile App für Android und iOS aus einem Guss programmiert, Elektor 11-12/2020: www.elektormagazine.de/200265-01
- [11] Verwenden von Beacons in RAD Studio: http://docwiki.embarcadero.com/RADStudio/Sydney/de/Verwenden\_von\_Beacons
- [12] RAD Studio Demos: https://github.com/Embarcadero/RADStudio10.3.3Demos
- [13] Map-Editor von RAD Studio: http://docwiki.embarcadero.com/IoT/en/BeaconFence\_Map\_Editor
- [14] iBeacons für Entwickler: https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf
- [15] Understanding the different types of BLE Beacons: https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/
- [16] AltBeacon-Spezifikationen: https://github.com/AltBeacon/spec
- [17] IoT Projects with Bluetooth Low Energy: www.oreilly.com/library/view/iot-projects-with/9781788399449/19495b89-8a0a-43f8-8dfd-955bdcc203db.xhtml
- [18] ixys L. Hernández-Rojas; Tiago M. Fernández-Caramés; Paula Fraga-Lamas und Carlos J. Escudero: Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications, in Sensors 201: www.mdpi.com/1424-8220/18/1/57
- [19] Beacons und Datenschutz: www.datenschutzkanzlei.de/fachbeitrag-beacons-gefahr-fuer-den-datenschutz/



# **C-Programmierung** auf Raspberry Pi

## Ein Beispielkapitel: Kommunizieren über WLAN

Von Dogan Ibrahim (Großbritannien)

Der Raspberry Pi wird traditionell mit Python programmiert. Obwohl Python eine sehr mächtige Sprache ist, ist unter Programmierern C wahrscheinlich die am häufigsten verwendete Sprache, nicht nur weil sie an fast allen technischen Hochschulen und Universitäten gelehrt wird. Dieser Artikel bietet einen Vorgeschmack auf C auf dem RPi, um Ihnen zu helfen, den Aufwand abzuschätzen, den Sie für die Entwicklung Ihrer eigenen hardwarebasierten Projekte betreiben müssen. Sie können von Ihrem Heimlabor aus arbeiten. mit einem Buch und Ihrem vertrauten RPi als Lehrmaterial. Let's C what happens!

Anmerkung der Redaktion: Dieser Artikel ist ein Auszug aus dem 376 Seiten starken Buch C Programming on Raspberry Pi - Develop innovative hardware-based projects in C, neu formatiert und leicht bearbeitet, um den Standards und dem Seitenlayout der Zeitschrift Elektor zu entsprechen. Da es sich um einen Auszug aus einer größeren Publikation handelt, können sich einige Begriffe in diesem Artikel auf Diskussionen an anderer Stelle des Buchs beziehen. Der Autor und die Redaktion haben jedoch ihr Bestes getan, um solche Fälle zu vermeiden, und helfen bei Rückfragen gerne weiter. Kontaktinformationen finden Sie im Kasten Fragen oder Kommentare?

#### **UDP und TCP/IP**

Die Kommunikation über eine WLAN-Verbindung erfolgt in Form von Client und Server. Zum Senden und Empfangen von Datenpaketen werden Sockets verwendet. Das Gerät auf der Serverseite wartet normalerweise auf eine Verbindung von den Clients, und sobald diese hergestellt ist, kann eine Zwei-Wege-Kommunikation beginnen. Für das Senden und Empfangen von Datenpaketen über eine WLAN-Verbindung werden hauptsächlich zwei Protokolle verwendet: UDP und TCP. TCP ist ein verbindungsbasiertes Protokoll, das die Zustellung von Paketen garantiert. Die Pakete werden mit Sequenznummern versehen und der Empfang aller Pakete wird bestätigt, um zu verhindern, dass sie in falscher Reihenfolge ankommen. Aufgrund dieser Bestätigungen ist



TCP in der Regel zwar langsam, aber zuverlässig, da es die Zustellung der Pakete garantiert. UDP hingegen ist nicht verbindungsbasiert. Die Pakete haben keine Sequenznummern und daher gibt es keine Garantie, dass sie an ihrem Ziel ankommen. UDP hat einen geringeren Overhead als TCP und ist daher schneller. In Tabelle 1 sind einige der Unterschiede zwischen den Protokollen TCP und UDP aufgeführt.

Tabelle 1. TCP- und UDP-Pake	tkommunikation
ТСР	UDP
Pakete haben Sequenznum- mern und die Zustellung jedes Pakets wird bestätigt	Es gibt keine Zustellungsbestätigung
Langsam	Schnell
Kein Paketverlust	Pakete können verloren gehen
Großer Overhead	Geringer Overhead
Benötigt mehr Ressourcen	Benötigt weniger Ressourcen
Verbindungsbasiert	Not connection-based
More difficult to program	Nicht verbindungsbasiert
Beispiele: HTTP, HTTPS, FTP	Beispiele: DNS, DHCP, Computerspiele





#### **UDP-Kommunikation**

Bild 1 zeigt die UDP-Kommunikation über eine WLAN-Verbindung.

#### Server

- 1. UDP-Socket erstellen
- 2. Binden des Sockets an die Serveradresse
- 3. Warten, bis das Datagramm-Paket vom Client eintrifft
- 4. Verarbeitung des Datagramm-Pakets
- 5. Senden einer Antwort an den Client oder Schließen des Sockets
- 6. Zurück zu Schritt 3 (falls nicht geschlossen)

#### Client

- 1. UDP-Socket erstellen
- 2. Senden einer Nachricht an den Server
- 3. Warten, bis eine Antwort vom Server eintrifft
- 4. Verarbeiten der Antwort
- 5. Zurück zu Schritt 2 oder Schließen des Sockets

#### Projekt 1: Kommunikation zwischen Raspberry Pi als Server mit einem Android-Smartphone über UDP

In diesem Projekt wird die UDP-Kommunikation verwendet, um Daten von und zu einem Android-Smartphone zu empfangen und zu senden. Der Raspberry Pi ist hier der Server und das Android-Smartphone der Client. **Ziel**: Dieses Projekt soll zeigen, wie eine UDP-Kommunikation zwischen einem Raspberry Pi und einem Android-Smartphone hergestellt werden kann.

**Blockdiagramm**: **Bild 2** zeigt die allgemeine Struktur des Projekts. Der Raspberry Pi und das Android-Smartphone kommunizieren über eine WLAN-Router-Verbindung.

**Programmlisting**: Das Programm *MyServer.c* ist in **Listing 1** dargestellt und kann aus dem Software-Support-Paket für das Buch entnommen werden. Gehen Sie zu [1], scrollen Sie nach unten zu *Downloads* und klicken Sie dann auf *Software\_C Programming on Raspberry Pi*. Speichern Sie die *.zip*-Datei lokal und extrahieren Sie dann entweder alle Dateien oder nur *MyServer.c*.

Zu Beginn von *MyServer.c* werden die erforderlichen Header-Dateien in das Programm eingebunden. Die Nachricht *Hello from Raspberry Pi* 

20:48 🗳 🖄			হি 🗐 41%	
÷			Q	÷
>_ UDP	UDP Ter mightyIT Contains ads	<b>minal</b> In-app purcha	ses	
Unins	tall	Ope	n	
Rate this a Tell others wha	<b>pp</b> t you think			
Bild 3. UDP-Tei Smartphone.	rminal-Apps	auf einem A	ndroid-	

JDP Terminal		
SSID: <unk< th=""><th>nown ssi</th><th>id&gt;</th></unk<>	nown ssi	id>
SSID: <unk< th=""><th>nown ssi</th><th>id&gt;</th></unk<>	nown ssi	id>
SSID: <unk< td=""><td>nown ssi</td><td>id&gt;</td></unk<>	nown ssi	id>
SSID: <unk Receive Packets 192.168.1.219</unk 	nown ssi C	i <b>d&gt;</b> 5000
SSID: <unk Receive Packets 192.168.1.219</unk 	nown ssi C	i <b>d&gt;</b>
SSID: <unk Receive Packets 192.168.1.219 Send Packets</unk 	nown ssi	i <b>d&gt;</b>
SSID: <unk Receive Packets 192.168.1.219 Send Packets O Broadcast O Unicast</unk 	nown ssi C	i <b>d&gt;</b>
SSID: <unk Receive Packets 192.168.1.219 Send Packets Broadcast © Unicast 192.168.1.202</unk 	nown ssi C	id> 5000 5000
SSID: <unk Receive Packets 192.168.1.219 Send Packets Broadcast © Unicast 192.168.1.202</unk 	nown ssi C	id> 5000 5000

20:47 🛛 🗢 –	≋al 41%≜
<unknown ssid=""> Receive at 192.168.1.219:5000</unknown>	ASCII
[31 Dec 2020 20:47:01] ASCII:	
(192.168.1.202:5000): Hello from Raspb	erry Pi 2
Auto Scroll	
A	This is Androi
ASCIE This is	another messag
Unicast Packets at 192.168.1.202:5000	_
PC SCREEN	rver MyServer

wird in einem Zeichenarray msg gespeichert. Anschließend wird ein UDP-Socket erstellt, indem die Funktion socket aufgerufen und das Handle in der Variablen sock gespeichert wird. Dann werden die Details des Server-Computers, also des Raspberry Pis angegeben, wobei die Adresse auf INADDR\_ANY gesetzt wird, so dass jeder andere Computer im selben Netzwerk mit der auf 5000 gesetzten Portnummer eine Kommunikation mit dem Raspberry Pi herstellen kann. Die Daten des Server-Computers werden dann durch den Aufruf der Funktion bind an den angegebenen Port gebunden. Der Rest des Programms läuft in einer Schleife ab. Innerhalb dieser Schleife wird die Funktion recvfrom aufgerufen, um auf den Empfang eines Datenpakets vom Client-Computer (Android-Smartphone) zu warten. Beachten Sie, dass dies ein blockierender Call ist und die Funktion wartet, bis Daten vom Client empfangen werden. Das Programm wird beendet, wenn das Zeichen x vom Client-Computer empfangen wird. Ein NULL-Zeichen wird zu den empfangenen Daten hinzugefügt und mit der printf-Funktion auf dem PC-Bildschirm des Raspberry Pi als String angezeigt. Eine Integer-Variable namens count wird in ein Zeichen umgewandelt und an das Ende des Zeichenarrays msg gehängt. Dies wird an den Client-Computer gesendet, so dass beim ersten Mal in der Schleife der Client-Computer Hello from Raspberry Pi1 anzeigt, beim zweiten Mal Hello from Raspberry Pi 2 und so weiter. Der Socket wird kurz vor Beendigung des Programms geschlossen. Das Programm kann wie folgt kompiliert und ausgeführt werden:

#### gcc -o MyServer MyServer.c sudo ./MyServer

Es gibt viele UDP-Apps, die im Play-Store kostenlos erhältlich sind. In diesem Projekt wird das UDP-Terminal von mightyIT verwendet (siehe Bild 3). Geben Sie die Portnummer und die IP-Adresse Ihres Android-Smartphones ein und klicken Sie auf Start Terminal, wie in Bild 4 gezeigt. Ein Beispiel für die Ausführung des Programms ist in Bild 5 zu sehen, wo sowohl der Smartphone- als auch der PC-Bildschirm abgebildet sind.

Hinweis: Sie können die IP-Adresse Ihres Raspberry Pi mit dem Befehl ipconfig herausfinden.

210346-02

#### Haben Sie Fragen oder Kommentare?

Haben Sie Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an den Autor unter d.ibrahim@btinternet.com oder an Elektor unter redaktion@elektor.de.

#### **Ein Beitrag von**

Text: Dogan Ibrahim Redaktion: Jan Buiting Übersetzung: Rolf Gerstendorf Layout: Harmen Heida

#### 

#### Listing 1. MyServer.c

```
RASPBERRY PI - ANDROID SMARTPHONE COMMUNICATION
This is an UDP program. The program receives and then sends
messages to a smartphone over the UDP socket. Program terminates
when character x is sent from the smartphone.
This is the UDP server program, communicating over port 5000.
Author: Dogan Ibrahim
File : MyServer.c
Date : December 2020
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#define Port 5000
#define BUFFSIZE 1024
// Start of MAIN program
11
int main(void)
{
    int sock, len, num, count = 1;
    char buffer[BUFFSIZE];
    struct sockaddr_in serveraddr, clientaddr;
    char msg[26] = "Hello from Raspberry Pi
    sock = socket(AF_INET, SOCK_DGRAM, 0);
    len = sizeof(clientaddr);
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = INADDR_ANY;
    serveraddr.sin_port = htons(Port);
    bind(sock,(struct sockaddr *)&serveraddr,
   sizeof(serveraddr));
    while(1)
    {
        num = recvfrom(sock, buffer, BUFFSIZE,MSG_WAITALL,
               (struct sockaddr *)&clientaddr, &len);
        if(buffer[0] == 'x')break;
        buffer[num] = '\0';
        printf("%s\n",buffer);
        sprintf(&msg[24], "%d",count);
        count++;
        sendto(sock, &msg, strlen(msg), MSG_CONFIRM,
               (struct sockaddr *)&clientaddr, len);
     close(sock);
```





- > Book: D. Ibrahim, C Programming on Raspberry Pi (Elektor 2021) (SKU 19703) www.elektor.de/19703
- > E-Book: D. Ibrahim, C Programming on Raspberry Pi (Elektor 2021) (SKU 19704) www.elektor.de/19704



# EMV-Vorkonformitätstest für Ihr DC-versorgtes Projekt

## Teil 2: Die Hardware und wie man sie benutzt

Von Robert Schillinger (Würth Elektronik) und Ton Giesberts (Elektor)



In unserer letzten Ausgabe haben wir das DC Line Impedance Stabilization Network (LISN) vorgestellt, das Elektor und Würth Elektronik im Elektor Store anbieten. Diesmal gehen wir näher auf die Hardware ein und zeigen, wie das LISN für die EMV-Vorprüfung von elektronischen Designs eingesetzt wird.

Eine Netznachbildung oder LISN, wie diese im Englischen abgekürzt wird, ist ein Tiefpassfilter, das in der Regel zwischen einer Wechsel- oder Gleichstromquelle und dem Prüfling platziert wird, um eine bekannte Impedanz zu erzeugen und einen Anschluss für die Messung des Hochfrequenzrauschens (HF) bereitzustellen. Sie isoliert auch unerwünschte HF-Signale von der Stromquelle. Darüber hinaus können Netznachbildungsgeräte zur Vorhersage leitungsgebundener Emissionen für Diagnose- und Vor-Konformitätsprüfungen verwendet werden [1].

Das hier vorgestellte Dual DC LISN wird zur Messung der leitungsgebundenen Störaussendung eines Prüfobjekts im Frequenzbereich von 150 kHz bis 200 MHz eingesetzt. Das Gerät basiert auf der Norm CISPR 25/ ISO 7637 für elektrische Systeme in Kraftfahrzeugen. Es misst die HF-Störungen solcher Systeme auf beiden Kanälen mit Sperrinduktivitäten von 5 µH. Dieser Wert ist charakteristisch für Netznachbildungen, die zum Testen von Automobilgeräten verwendet werden. Der Entwurf basiert auf der Application Note DC2130A von Analog Devices (früher Linear Technology [2]). Der Anwender kann mit diesem Gerät das Rauschen im differenziellen Modus (DM) und das Gleichtaktrauschen (CM) messen.

#### **Die Schaltung der** Netznachbildung

Der obere Teil von **Bild 1** zeigt das Grundschema für alle LISN-Konstruktionen: eine HF-Induktivität zur Rauschunterdrückung. einen Kondensator zur Unterdrückung von Gleichstrom am HF-Ausgang und eine Impedanz von 50  $\Omega$  für diesen Messanschluss. In der Praxis ist diese letztgenannte Impedanz viel komplexer als ein einfacher Widerstand. Der untere Teil dieses Schaltplans zeigt, dass die Impedanz in drei Stufen unterteilt werden kann: Vordämpfung, Hochpassfilter, Nachdämpfung. Wir werden uns alle drei Abteilungen genauer ansehen. Die dargestellten Diagramme sind das Ergebnis von Computersimulationen









Bild 2. Eigenschaften der Netznachbildung.





Bild 4. Das vollständige Schaltbild unserer Netznachbildung.

#### EMC-Webinar: How to Debug a Flyback Design!

der Schaltungen.

Das Dämpfungsnetzwerk wird im Folgenden ohne die Koppelkondensatoren C3/C4 und C11/C12 beschrieben, die die untere Frequenz des Messbereichs auf die üblichen 150 kHz begrenzen (Grenzfrequenz liegt bei 64 kHz, bei 125 kHz beträgt die Dämpfung etwa -1 dB).

#### Vordämpfung:

In der ersten Stufe durchläuft das Rauschen ein Pi-Filter [V(-3db)] mit -3 dB an 50  $\Omega$ , um die Rauschspannung zu verringern. Die blaue Kurve in **Bild 2** zeigt die Dämpfung über der Frequenz.

#### Hochpassfilter:

Die zweite Stufe ist ein Hochpassfilter dritter Ordnung [V(hpf)] mit einer -3-dB-Eckfrequenz von 5,5 kHz, um den Frequenzbereich zwischen 9 kHz und etwa 200 MHz zu begrenzen, wie in Bild 2 (rote Kurve) dargestellt. Diese Stufe filtert nur Frequenzen unter 6 kHz und ist nicht unbedingt erforderlich, um die 10-dB-Dämpfung des Signals zu erreichen. Man könnte sie auch aus der Schaltung entfernen.

#### Nachdämpfung:

In der letzten Stufe wird die Dämpfung auf -10 dB [V(out)] (türkisfarbene Kurve in Bild 2) gesenkt. Dies ist ein typischer Wert für eine Netznachbildung und muss durch die Messausrüstung (zum Beispiel interne 10-dB-Verstärkung) oder durch Nachbearbeitung der Messdaten in einer Excel-Berechnung korrigiert werden.

Die 5-µH-Induktivitäten blockieren den Weg zurück zur Stromversorgung für das vom Prüfling erzeugte Rauschen über den gesamten Frequenzbereich. 5 µH ist ein typischer Wert, der in CISPR25 verwendet wird. Bei einem CISPR22-Aufbau sollten Sie sich Frequenzen unter 400 kHz ansehen. Aufgrund der niedrigen Induktivität (abnehmende Wechselstromunterdrückung bei niedrigeren Frequenzen) werden Sie niedrigere Pegel messen.

Die abnehmende Impedanz über 100 MHz (Bild 3) resultiert aus der parasitären Kapazität der Klemmdioden. Diese Dioden sind wichtig, um hohe Spannungsspitzen an den HF-Ausgängen zu verhindern. Wenn Sie ein Oszilloskop mit wenig anfälligen und ohnehin intern geschützten Eingängen für Pre-Compliance-Tests verwenden, können die Dioden weggelassen werden.

die Grundlagen der Fehlersuche in einer Sperrwandlerschaltung mit Hilfe eines Oszilloskops und erörtert, welche zusätzliche Ausrüstung erforderlich ist. Wir geben Ihnen einen tieferen Einblick in geschaltete Stromversorungen, ihre Rauschquellen und die Art der Filterung, die erforderlich ist, um EMV-Konformitätstests erfolgreich zu bestehen. Am Ende des Webinars sollten Sie in der Lage sein, zwischen DM-Rauschen (differential mode) und scheiden und zu verstehen, wie man den dominanten Modus bestimmt.

Dieses Webinar findet am 2. Dezember 2021 um 16:00 Uhr MEZ statt. Weitere Infor-

Bild 4 zeigt das vollständige Schaltbild unseres Dual DC LISN. Die beschriebenen HF-Ausgangsschaltungen sind leicht zu erkennen, wobei X1 und X8 die HF-Ausgänge für die Messungen der leitungsgebundenen Störungen sind. Die DC-Stromversorgung (Eingang, Einzelspannung plus zusätzliche Masseverbindung zur Stromversorgung und zu GND) ist an X2, X5 und X7 angeschlossen, die Stromversorgung des Prüflings (Ausgang) an X3 und X6 (oder an X4. der nicht im LISN-Bausatz enthalten ist).

Die normalerweise für diese LISN vorgeschriebenen 5-µH-Induktivitäten werden in diesem Entwurf durch fünf in Reihe geschaltete Standardinduktivitäten von 1,1 µH aus dem Produktprogramm von Würth Elektronik realisiert. Diese Induktivitäten wurden aufgrund des maximalen Stroms von 10 A in Verbindung mit ihrer kleinen Bauform, der hohen Resonanzfrequenz und damit geringer parasitärer Kapazität ausgewählt, was zu einem guten Induktivitäts-Strom-Verhältnis und damit geringen Induktivitätsabfall über den Strombereich bis 10 A führt. In den meisten handelsüblichen Netznachbildungen werden zu diesem Zweck Luftspulen eingesetzt, die aber deutlich mehr Platz benötigen und zudem nicht als Standardbauteile erhältlich sind

#### Kaufen oder selbst bauen?

Ein kompletter Bausatz für diese Netznachbildung ist, gesponsert von Würth Elektronik, im Elektor-Shop [3] erhältlich. Da alle SMD-Bauteile bereits vormontiert sind, müssen Sie nur noch die Steckverbinder anlöten und die Platine in das vorgebohrte und gefräste Gehäuse einsetzen, um das Messgerät fertigzustellen.

Für die ganz Mutigen unter Ihnen finden sich die Gerber-Dateien, die Sie benötigen, um Ihre eigene vierlagige Platine zu bestellen, auf der Elektor-Lab-Seite dieses Projekts [4] (siehe **Bild 5**). Die detaillierte Stückliste hilft Ihnen bei der Beschaffung der Bauteile für den Bau Ihres eigenen LISN.

Die SMD-Bauteile auf der Oberseite der Platine, C5...C10 und L2...L11, können



Bild 5. Das Platinenlayout.



in einem Reflow-Ofen gelötet werden. Das Löten der Spulen mit einer Heißluft-Rework-Station oder einem Lötkolben mit feiner Spitze dürfte schwierig bis unmöglich sein, da die Pads zwischen den Spulen fast unzugänglich sind. Außerdem erfordert die vierlagige Platine eine Menge Lötleistung, um ein Pad aufzuheizen, vor allem, wenn es mit einer Versorgungsebene verbunden ist. Bauteile auf der Unterseite der Platine wie die kleinen 0603-Widerstände. Kondensatoren und zwei kleinere Induktivitäten können mit Heißluft gelötet werden. Normalerweise lässt sich ein kleiner 0603-Widerstand auch mit einem kleinen Lötkolben löten, aber es fließt doch sehr viel Wärme in die vier Schichten. Wenn Sie Heißluft verwenden, stellen Sie den Luftstrom auf ein Minimum ein, sonst driften Ihnen diese kleinen Bauteile über die Platinenoberfläche.

Die SMA-Buchsen (X1, X8) können ebenfalls mit Heißluft gelötet werden. Tragen Sie zunächst etwas Lötpaste auf alle Pads auf der Ober- und Unterseite der ersten Buchse auf, stecken Sie die SMA-Buchse auf die Leiterplatte und stellen Sie den Luftstrom höher ein, um die Buchse auf beiden Seiten zu verlöten. Löten Sie zuerst den Mittelstift, dann die Massekontakte auf beiden Seiten der Platine. Achten Sie darauf, dass der Mittelstift in der Mitte des entsprechenden Pads sitzt.

Die Federkontakte (Kontaktfinger) können auch mit Heißluft und Lötpaste gelötet werden. Geben Sie etwas Paste auf die Pads (X13...X16) und stellen Sie den Kontakt aufrecht hin. Hören Sie auf, wenn die gesamte Lötpaste fließt und sich der Kontakt deutlich sichtbar setzt.

Die Dioden im MiniMELF-Gehäuse können mit einem Lötkolben mit kleiner Meißel-

spitze und sehr dünnem Lötdraht von zum Beispiel 0,35 mm gelötet werden. Geben Sie zunächst etwas Lötzinn auf ein Pad, vorzugsweise auf das Masse-Pad, da es eine Weile dauert, bis das Lot auf diesen Pads fließt. Das Pad auf der Signalseite ist leichter zu löten. Wenn Sie fertig sind, können Sie den MiniMELF platzieren. Richten Sie den MiniMELF mit einer Pinzette an der Grundfläche der Platine aus und verlöten Sie ein Ende. Schauen Sie genau hin, um sicherzustellen, dass die Diode in der Mitte des vorgesehenen Bereichs auf der Leiterplatte platziert ist. Bei jedem Diodenpaar werden eine Kathode (schwarzer Ring) und die Anode der anderen Diode an Masse gelötet. Löten Sie nun das andere Ende an und verlöten Sie die erste Lötstelle erneut mit etwas zusätzlichem Lötzinn.

Wenn alle Teile auf beiden Seiten auf der Platine angebracht sind, befestigen Sie die Sicherheits-Bananenbuchsen am Gehäuse, wobei die Kontakte senkrecht stehen müssen. Schieben Sie die Platine in das Gehäuse, wobei die SMA-Buchsen in die entsprechenden Löcher gesteckt werden. Achten Sie darauf, dass die Federkontakte an der Unterseite der Leiterplatte nicht durch den Gehäuserand verbogen werden. Üben Sie etwas Druck auf die Leiterplatte aus und setzen Sie die Unterlegscheibe (Sicherungsring) und die Mutter auf die SMA-Anschlüsse. Löten Sie dann die Kontakte der Bananenstecker an die Pads (X2, X3, X5, X6, X7), während Sie die Platine noch festhalten. Setzen Sie noch den Deckel auf das Gehäuse. dann ist die Netznachbildung einsatzbereit!

#### Verwendung der Netznachbildung

In Bild 6 sind die Spannungen und Ströme dargestellt, die bei der Messung der beiden Arten von leitungsgebundenem Rauschen auftreten. Bei Differenz-Rauschen fließt der Strom von der Quelle zur Last und auf der anderen Leitung wieder zurück. Die Erdung ist in diesem Fall nicht wichtig. Die Gegentaktspannung wird durch  $V_{dm} = 0.5^* (V_p - V_p)$ berechnet.

Beim Gleichtakt-Rauschen fließt der Strom auf beiden Leitungen in die gleiche Richtung und kehrt über die Erde zur Quelle zurück. Dieser Rückstrom kann auch durch parasitäre Kopplungseffekte geleitet werden. Die Gleichtaktspannung wird mit  $V_{cm} = 0.5^*(V_p + V_p)$  berechnet.

Es gibt zwei Möglichkeiten, das Differenzund Gleichtaktrauschen mit unserem LISN zu messen (siehe **Bild 7**). Bei der ersten wird ein so genannter Combiner von Mini-Circuits [5] verwendet, um beide HF-Ausgänge zu addieren (entweder mit oder ohne 180°-Phasenverschiebung) und anschließend die Messungen und Berechnungen für DM/CM durchzuführen.

Der zweite Ansatz verwendet ein zweikanaliges digitales Oszilloskop und seine mathematischen Fähigkeiten (Addition und FFT). In diesem Fall werden die Combiner von Mini-Circuits nicht benötigt, da V1 und V2 direkt an die Eingangskanäle des Oszilloskops angeschlossen werden.

Schließen Sie das Oszilloskop oder den Analyzer vorsichtshalber nur dann an, wenn die an der Netznachbildung angeschlossene Spannungsversorgung eingeschaltet ist, um das Messgerät zu schützen. Trennen Sie den





Bild 8. Aufbau der Rauschmessung im Differenz-Modus.













Bild 10a. Erstes Layout, nicht konform.

Bild 10b. Verbesserte Anordnung und Filterung, CISPR22B-konform.

Bild 10. Zwei Platinenversionen eines Schaltnetzteils:



Bild 11. Rauschen im Vergleich zur Frequenz, beide Platinenversionen:

#### Technische Eigenschaften

Analysator, bevor Sie die Stromquelle zum LISN ausschalten oder abziehen. Schließen Sie alle nicht verwendeten Ausgänge mit 50  $\Omega$  ab.

Bild 8 zeigt, wie Rauschmessungen im Differenzmodus durchgeführt werden. Die beiden Ausgangsspannungen der LISN werden addiert, wobei eine zuvor um 180° phasenverschoben wurde. Die Gleichtaktspannungen heben sich im Idealfall auf, und der Pegel der Gegentaktspannung wird verdoppelt.

Um den richtigen Pegel für diese Messung zu erhalten, müssen die 6 dB vom gemessenen Wert abgezogen werden. Da die Combiner um -3 dB dämpfen, muss auch dies in die Rechnung mit einbezogen werden, um den korrekten Pegel zu erhalten. Zusammen mit den -10 dB der Netznachbildung müssen wir die Messungen also wie folgt korrigieren:

- > −6 dB + 3 dB + 10 dB = +7 dB (bei Verwendung des Combiners)
- > -6 dB + 10 dB = +4 dB (bei Mathematik-Funktionen des Oszilloskops)

In Bild 9 ist die Messung des Gleichtakt-Rauschens dargestellt. Im Idealfall ergibt die Summierung von V1 und V2 die doppelte Gleichtaktspannung, während sich die Gegentaktspannung aufhebt. Das bedeutet, dass bei gleich hohen Gleichtaktspannungen und exakt gleicher Phase auf beiden Leitern die Spannung doppelt (6 dB) so hoch ist wie auf den beiden Leitungen.

#### **Beispiel: Prüfung eines Schaltnetzteils**

Um zu zeigen, wie Pre-Compliance-Tests mit dieser Netznachbildung durchgeführt werden und wie die Ergebnisse verarbeitet werden sollten, wurden Messungen an zwei Beispielschaltungen vorgenommen. Die eine Platine in **Bild 10a** ist ein erstes grobes Lavout eines Schaltnetzteils. Die zweite Platine in Bild 10b ist die verbesserte Version derselben Schaltung nach einer Vorprüfung in einem professionellen EMV-Prüflabor. Anhand dieser beiden Messungen derselben Schaltung in einem professionellen Labor konnten wir die mit unserer eigenen Netznachbildung und der Messausrüstung in unserem Labor gemessenen EMV-Werte korrigieren. Dies ist eine gute Lösung für Pre-Complianceoder Vergleichsmessungen, um im eigenen Labor die Fehlersuche in einem Entwurf durchzuführen.

HF-Ptad		
Kanäle:	2 (mit Klemmdioden)	
Bandbreite:	200 MHz	
Induktivität:	5 μΗ    50 Ω	
Interne Abschwächung:	10 dB	
DC-Pfad		
Max. Strom:	< 10 ADC	
Max. Spannung:	< 60 VDC	
DC-Widerstand:	< 2 x 70 mO	

#### Leitungsgebundenes Rauschen

Es gibt zwei Arten von Leitungsrauschen, die beide mit diesem LISN gemessen werden können.

#### **Differenz-Modus**

D

Platinengröße:



wird durch ein hohes dl/dt verursacht; Schaltkreise wirken wie Schleifenantennen oder magnetische Dipole.

- > Strompfad wie im Schaltplan; relativ leicht zu erkennen
- > Kleiner Rückstrompfad
- > Relativ hohe Ströme
- > dI/dt ist der dominierende Faktor
- > Kann mit LC-, π-, T-Topologien gefiltert werd

#### **Gleichtakt-Modus**



wird durch hohe  $\Delta V/\Delta t$  verursacht; Schaltungen wirken wie elektrische Dipol- und Monopolantennen.

- > Unerwartete parasitäre Strompfade
- > Großer Rückstrompfad
- Relativ kleine Ströme (µA-Bereich)
- > dV/dt ist der dominierende Faktor
- > Kann mit stromkompensierten Drosseln und Y-Kondensatoren gefiltert werden



Die Messergebnisse der Gleichtakt- und Gegentakt-Leitungsstörungen dieses Tests sind in **Bild 11** dargestellt. Für beide Fälle hält der zweite Entwurf mit verbessertem Platinenlayout und zusätzlicher Filterung die Grenzwerte von CISPR22B ein.

Beide Kurven wurden von 150 kHz bis 30 MHz gemessen, dem Frequenzbereich, der für die Einhaltung von CISPR22 von Interesse ist. Bei höheren Frequenzen wurden die abgestrahlten Emissionen mit Antennen ermittelt. Die orangefarbenen Kurven stellen die Platine ohne Filter dar. Bei 650 kHz sehen Sie die Grundfrequenz des Schaltnetzteils mit 72 dBµV. Sie können auch einige Oberwellen bei 1,3 MHz, 1,95 MHz und so weiter erkennen. Die blaue Linie zeigt den Grenzwert der CISPR22-Klasse B an. Das leitungsgebundene Rauschen sollte etwa 10 dB unter dieser Linie liegen. Die hohe Spitze bei der Schaltfrequenz (650 kHz) des Netzteils musste deshalb um 22 dB reduziert werden!

Durch die Verbesserung des Layouts und das Hinzufügen von Filtern im zweiten Entwurf wurde die geforderte Verringerung dieser Spitze erreicht. Die gelben Kurven zeigen das neue Ergebnis von 45 dBµV bei 650 kHz.

Durch den Vergleich der Ergebnisse von EMV-Prüfungen in einem zertifizierten Labor mit denen unserer Netznachbildung kann der Messaufbau so angepasst werden, dass Ihre eigenen Tests einen guten Hinweis auf die EMV-Konformität Ihrer elektronischen Produkte liefern.

210373-02

#### PASSENDE PRODUKTE

- > Dual DC LISN kit (SKU 19869) www.elektor.de/19869
- STEMlab 125-14 Starter Kit (SKU 17939) www.elektor.de/17939

#### **Ein Beitrag von**

Idee und Design: Robert Schillinger (Würth Elektronik) Text: Robert Schillinger, Ton Giesberts Illustrationen: Robert Schillinger, Patrick Wielders Redaktion: Luc Lemmens, Stuart Cording Übersetzung: Rolf Gerstendorf Layout: Harmen Heida

#### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Senden Sie eine E-Mail an die Redaktion unter luc.lemmens@elektor.com oder kontaktieren Sie Elektor unter editor@elektor.com.

[1]

[2]

[3]

[4]

[5]

### 

#### STÜCKLISTE

WEBLINKS

Widerstände: (alle SMD 0603, 1%) R1,R13 = 820 Ω R2,R14 = 180 Ω R3,R4,R15,R16 = 680 Ω R5,R17 = 12 Ω R6,R18 = 27 Ω R7,R19 = 18 Ω R8,R20 = 910 Ω R9,R21 = 270 Ω R10,R22 = 220 Ω R11,R12,R23,R24 = 510 Ω

#### Kondensatoren:

- C1,C2,C13,C14 = 470 n, 50 V, SMD 0805, X7R (Würth Elektronik 885012207102)
- C3,C4,C11,C12 = 100 n, 250 V, SMD 1206, X7R (Würth Elektronik 885342208004)
- $$\label{eq:c5,C8} \begin{split} & \text{C5,C8} = 22 \; \mu, 100 \; \text{V}, \; \text{SMD WCAP-ASLL (8x10,5)}, \\ & (\text{Würth Elektronik 865060853003 oder} \\ & \text{865080853006)} \end{split}$$

68 November/Dezember 2021 www.elektormagazine.de

C6,C7,C9,C10 = 2,2 μ, 100 V, SMD 1210, X7R (Würth Elektronik 885012209071)

#### Induktivitäten:

Wikipedia über Netznachbildung: https://de.wikipedia.org/wiki/Netznachbildung

Application Note DC2130A von Analog Devices: https://bit.ly/2SST8dc

Projekt bei Elektor Labs: https://bit.ly/3BVaNSD Combiner von Mini-Circuits: https://bit.ly/3CrVzWg

Bausatz Dual DC LISN (Webadresse anpassen!): www.elektor.com/19869

L1,L12 = 1 m, 200 mA, SMD WE-LQS 5040 (Würth Elektronik 74404054102) L2...L11 = 1µ1, 15000 mA, SMD WE-HCI 7050 (Würth Elektronik 744314110)

#### Halbleiter:

D1...D8 = LL4148, 100 V, 100 mA, SMD MiniMelf (SOD-80)

D9,D10 = TVS-Diode, nicht verwendet/montiert

#### Außerdem:

- X1,X8 = Buchse SMA 9.52, 5 Stifte 180° (End Launch), für 1,6-mm-Platine (Würth Elektronik 60312242114510)
- X2,X3 = Sicherheits-Laboreinbaubuchse, 4 mm, für Gehäuseeinbau, 24 A, 1 kV, vergoldete

Kontakte, rot (Staubli 23.3000-22)

- X4 = nicht montiert
- X5,X6 = Sicherheits-Laboreinbaubuchse, 4 mm, für Gehäuseeinbau, 24 A, 1 kV, vergoldete Kontakte, blau (Staubli 23.3000-23)

X7 = Sicherheits-Laboreinbaubuchse, 4 mm, für Gehäuseeinbau, 24 A, 1 kV, vergoldete Kontakte, grün/gelb (Staubli 23.3000-20)

X13,X14,X15,X16 = EMV-Kontaktfeder, WE-SECF-Reihe, SMD, 7x2.5x13mm (Würth Elektronik 331161702513)

Gehäuse = 1590N1, Aluminium-Druckguss-Gehäuse aus der Hammond-Mfg-Reihe, 121,2 mm x 65,5 mm x 39,8 mm

#### Platine 210296-1 v1.0

#### **Optional:**

- (zum Anschluss des LISN an das STEMlab 250-14)
- 4 SMA-Stecker BNC-Adapter bei Verwendung von zwei Standard-BNC-Kabeln oder zwei SMA-Kabeln (Stecker-Stecker)

# Parallax Propeller 2

Teil 5: Das Innenleben des Smart Pins

#### Von Mathias Claußen (Elektor)

Im letzten Teil dieser Artikelserie werden wir die Smart-Pin-Funktion vorstellen, die eine universelle und flexible Konfigurationen der I/O-Pins ermöglicht. Außerdem schauen wir uns die Pull-up- und Pull-down-Widerstände der I/O-Pins genauer an. Aber es gibt noch mehr zu entdecken.

Da die Pins des Propeller 2 mehr als die bei anderen Mikrocontrollern üblichen Funktionen haben, offenbart ein genauerer Blick einige interessante Einblicke. Als Referenz schauen wir uns zunächst die GPIO-Struktur eines Atmega328P von Microchip Technology an. Danach wenden wir uns dem Propeller 2 zu, um zu sehen, ob es Unterschiede gibt und wie sich diese als nützlich erweisen könnten. Am Ende haben wir leichtes Spiel, ein paar Tastenzustände auszulesen.

#### Wie I/O-Pins anderer MCUs funktionieren

Wer mit dem ATmega328P vertraut ist, weiß, dass es grundsätzlich vier Zustände für I/O-Pins gibt: Eingang, Eingang mit Pullup, Ausgang mit Low und Ausgang mit High. Analoge Funktionen sind eine Spezialität nur weniger Pins des ATmega328P und werden hier nicht behandelt. Im Datenblatt-Ausschnitt in **Bild 1** können wir sehen, wie ein I/O-Pin grundsätzlich aufgebaut ist. In rot sind einige interessante Abschnitte bezeichnet. Wir sehen im Wesentlichen einen Steuermechanismus für einen Pull-up-Widerstand, der mit 20 k $\Omega$  bis 50 k $\Omega$  spezifiziert ist, bestehend aus einem FET und dem Widerstand selbst, sowie einer Steuerlogik zur Aktivierung des FET (wie der Teil in **Bild 2** zeigt). Im unteren Teil von Bild 1 ist ein Übertragungsgatter (Transmission Gate) zu sehen (**Bild 3**), das als digital gesteuerter Schalter fungiert und analoge Spannungen durchlässt.

Interessant ist das *SLEEP*-Signal, das das Übertragungsgatter deaktiviert und dessen Ausgang gleichzeitig mit einem speziellen FET an Masse bindet. Der Grund dafür ist der Schmitt-Trigger, der sich am Ausgang des Transmissionsgatters befindet. Er hat die Aufgabe, einen analogen Spannungspegel in einen binären Wert von Null oder Eins umzuwandeln. In der Mitte von **Bild 4** befindet sich die Ausgangsstufe mit Freigabe (Enable). Ohne Freigabe wird die Ausgangsstufe abgeschaltet, oder sie steuert einen Low- oder High-Pegel, je nach dem angelegten Eingang. Viel Aufwand für bloßes Ein- und Ausschalten, aber es zeigt die Grundlagen, und der Pullup ist aus einem Grund sehr







Bild 2. Atmega328: FET für Pull-up.



Bild 3. Transmission Gate.



Bild 4. Ausgangspfad des ATmega328.

nützlich. Wir können nur global alle Pins auf einmal in den Ruhezustand versetzen (wie andere Teile des ATmega328). Wenn wir aber einen Pin unbenutzt lassen und nichts damit tun wollen, schwebt sein Eingang, was bedeutet, dass er zufälliges Rauschen aufnimmt und der Schmitt-Trigger dies in binäre Nullen oder Einsen umwandelt. Da das Rauschen zufällig ist, schalten der Schmitt-Trigger und die interne Logik permanent und sehr schnell zwischen Null und Eins um. Jedes Umschalten verschwendet aber ein bisschen Energie, was nicht wünschenswert ist. Dieses Verhalten sollte vor allem, wenn der Controller mit Batteriestrom betreiben wird, vermieden werden. Durch den Pull-up wird die Spannung am Eingang fest auf VCC gesetzt und ein zufälliges Schalten des Eingangs verhindert. Was aber macht der Propeller 2 in dieser Hinsicht anders?

#### **Smart Pins am Propeller 2**

Wie bereits erwähnt, gibt es beim Propeller 2 keine einfachen I/O-Pins. Selbst wenn es nur um die grundlegenden Ein- und Ausgangsfunktionen geht, zeigt ein Blick in das vorläufige Datenblatt des Propeller 2, dass wir mehr als vier Auswahlmöglichkeiten haben, denn alle Pins können im digitalen oder analogen Eingangs- oder Ausgangsmodus arbeiten.

Wir beginnen mit einem digitalen Eingang. Beim ATmega328 war alles ganz einfach: ein Transmissionsgatter und ein Schmitt-Trigger zum Lesen des Signals. Der Propeller 2 bietet ein bisschen mehr Intelligenz im Inneren des Pins. **Bild 5** zeigt den digitalen Eingangspfad, oder besser gesagt die Pfade für einen einzigen Pin. Das erste, was auffällt, sind zwei Eingangsselektoren zur Auswahl. So können für jeden dieser Eingänge der aktuelle Pin und drei Pins (aufwärts oder abwärts) daneben ausgewählt werden. Nach dieser Auswahl können wir das invertierte oder nicht invertierte Signal verwenden, was zu einem A- oder B-Term führt. Die A- und B-Terme treffen dann auf einen zweiten Teil, in dem logische Operationen oder Filterungen durchgeführt werden können. Das Endergebnis dieser Bearbeitung wird dem System als IN-Signal zur Verfügung gestellt. Das macht die Standard-Pins etwas komplizierter, aber auch deutlich vielseitiger als die eines Atmega328P.

Beim digitalen Ausgang sind die Dinge einfach: Wir haben es nur mit einem Low- und einem High-Ausgangsignal zu tun, auf den ersten Blick also nichts Ausgefallenes. Doch was ist mit dem dem



Bild 5. Eingangspfade des I/O-Pin beim Propeller 2.

Pull-up-Widerstand? Der Propeller 2 besitzt mehr als nur einen und kann sowohl Pull-up und Pull-down. Die Kombinationen sind recht einfach:

- **>** 1,5 kΩ
- **>** 15 kΩ
- > 150 kΩ
  > 19 Ω
- > 1 mA
- **>** 100 µA
- > 10 µA
- > Float

Da wir wählen können, ob wir Pull-up- oder Pull-down-Widerstände oder Stromquellen haben wollen, macht dies die Pins sehr flexibel, auch für die verschiedenen Busse geeignet, die wir benötigen oder mit denen wir interagieren wollen.

Die Pull-Ups oder Pull-Downs werden nicht wie beim ATmega328 mit einem separaten FET ein- oder ausgeschaltet; stattdessen bestimmen diese Widerstände als Ansteuerungsstärke (Drive Strength), als ob sie zwischen der Treiberstufe und dem Pin-Ausgang stünden. Die Ansteuerungsstärke eines I/O gibt an, wie viel Strom man maximal treiben und ableiten kann. Um den Pull-Up oder Pull-Down zu verwenden, schalten Sie den Pin als Ausgang mit dem Widerstand und verwenden Low oder High für Pull-Up oder Pull-Down. Aber das ist noch nicht alles: Bild 6 gibt einen Einblick in das Low-Level-Design eines Pins. Wir haben auch einen DAC und ADC für jeden einzelnen Pin, der auch in einem analogen Modus verwendet werden kann. Wie ich bereits früher in dieser Artikelreihe erwähnt habe, ist die Dokumentation noch nicht vollständig, so dass einige Details für bestimmte I/O-Konfigurationen fehlen, insbesondere wenn es um die Konfigurationen auf der untersten Ebene geht. Dies ist bei noch experimentellen Controllern noch vor der Serienreife aber nichts Außergewöhnliches. Bild 7 erlaubt Ihnen einen Überblick über das Smart-Pin-Konfigurationsregister. Aber genug der Theorie für den Moment.

#### Praktische Anwendung

Für unseren ersten Praxistest verwenden wir unsere vier Eingabetasten. Wir können eine LED mit einem Tastendruck zum Leuchten bringen. Dazu erweitern wir unseren bestehenden SPIN2-Code und nutzen all



Bild 6. Low-Level-Beschreibung des Propeller-2-Pins.



Bild 7. Smart-Pin-Konfigurationsregister.

die netten Fähigkeiten wie beispielsweise die Steuerung eines I/O-Pins für eine LED und unsere prints()-Funktion zum Schreiben von Strings an einen UART. Das sollte sehr einfach sein. Wir lesen den Zustand des Pins und geben ihn an die LED weiter, und zwar viermal, einmal für jede LED und jede Taste. Wenn eine Taste gedrückt oder losgelassen wird, geben wir außerdem einen entsprechenden String aus. Beginnen wir mit der Eingangskonfiguration. Da die Hardware-Tasten keine eigenen Pull-up/down-Widerstände haben, kommen die internen Widerstände des Chips zum Einsatz. Aus Bild 7 und Bild 8 müssen wir den Wert für das Konfigurationsregister berechnen. Da wir mit Blick auf Bild 5 nur den logischen Pegel unseres Pins haben wollen, wählen wir den gleichen, nicht invertierten Eingang A und B und den Ausgang A ohne jegliche Logik oder Filterung. Aus dem Propeller-2-Datenblatt ergibt sich der Registerwert 0000 0000 000 000 010 010 00 00000 in binärer Form oder 0x900 in hexadezimaler Form, wenn wir einen 15-k $\Omega$ -Pull-up verwenden wollen.

#### **WEBLINK**

[1] Elektor auf GitHub: https://github.com/ElektorLabs/200479-propeller2-hands-on

Mode Groups	%PPPPPPPPPPP	Input	Output	Notes	CIOHHHL	LL Bits Explained	
Non-ADC,	0000_CIOHHHLLL	PinA Logic	OUT	Output enabled by	Clocked	Clocked mode I/O	
Non-DAC Modes	0001_CIOHHHLLL	PinA Logic	IN	floating)	avallab	le with C:	
	0010_CIOHHHLLL	PinB Logic	IN		С	IN/OUT	
	0011_CIOHHHLLL	PinA Schmitt	OUT		0	Live	
	0100_CIOHHHLLL	PinA Schmitt	IN		1	Clocked	
	0101_CIOHHHLLL	PinB Schmitt	IN		Can inv	Can invert polarity of	
	0110_CIOHHHLLL	PinA > PinB	OUT		1/0 wit	n 1, 0:	
	0111_CIOHHHLLL	PinA > PinB	IN		I	IN	
ADC Modes	100000_0HHHLLL	ADC, GIO 1x	OUT	Output enabled by	0	Non-Inverted	
without DAC	100001_0HHHLLL	ADC, VIO 1x	OUT	<pre>DIR = 1 (otherwise floating)</pre>	1	Inverted	
	100010_0HHHLLL	ADC, PinB 1x	OUT		0	ουτ	
	100011_0HHHLLL	ADC, PinA 1x	OUT		0	Non-Inverted	
	100100_0HHHLLL	ADC, PinA 3.16x	OUT		1	Inverted	
	100101_0HHHLLL	ADC, PinA 10x	OUT		Pull un	Pull up and/or down with	
	100110_0HHHLLL	ADC, PinA 31.6x	OUT		HHH/LLL	:	
	100111_0HHHLLL	ADC, PinA 100x	OUT		HHH/L	LL Drive	
ADC + DAC	10100_DDDDDDDD	ADC, PINA 1x	DAC 990 Q, 3.3 V	ADC activated when OUT = 1 DDDDDDDDD = DAC Level	000	Fast	
Modes	10101_DDDDDDDD	ADC, PINA 1x	DAC 600 Ω, 2.0 V		001	1.5 kΩ	
	10110_DDDDDDDD	ADC, PINA 1x	DAC 123.75 Q, 3.3 V		010	15 kΩ	
	10111_DDDDDDDD	ADC, PINA 1x	DAC 75 Ω, 2.0 V		011	150 kΩ	
DAC	1100_CDDDDDDDD	PinA > D	OUT, 1.5 kΩ	HHH=001, LLL=001 DDDDDDDD = DAC Level Output enabled by DIR = 1 (otherwise floating)	100	1000 µA	
Comparison Modes (Non-	1101_CDDDDDDDD	PinA > D	!IN, 1.5 kΩ		101	100 µA	
ADC)	1110_CDDDDDDDD	PinB > D	IN, 1.5 kΩ		110	10 µA	
	1111_CDDDDDDDD	PinB > D	!IN, 1.5 kΩ		111	Float	

Bild 8. Übersicht der Pin-Modi beim Propeller 2.

Wir fügen nun vier Eingänge und vier Ausgänge auf unserer kleinen Zusatzplatine (Bild 9) zum Propeller 2 Evaluation Board hinzu. In Listing 1 sehen Sie den modifizierten SPIN2-Code zum Einrichten unserer vier Eingangspins mit Pull-ups und die Ausgangskonfiguration für unsere LED-Ausgangspins. Wir verwenden vier globale Variablen, um den zuletzt gelesenen Pinpegel zu speichern und nur dann eine serielle Nachricht zu senden, wenn sich ein Pinzustand geändert hat. Wir erweitern den Code um zwei Funktionen, die erste zur Initialisierung der LEDs und die zweite zur Initialisierung unserer Eingänge. In Listing 2 sehen Sie, dass vier Bytes deklariert wurden, um den Zustand der vier angeschlossenen Schalter zu speichern. Sie können auch sehen, dass der Code mit den beiden Funktionen für die Initialisierung geändert wurde (Listing 3). Der Trick, um die Eingänge mit aktiver Pull-up-Fähigkeit zu betreiben, ist in den Bits in Listing 4 versteckt. Wir stellen die Ein- und Ausgänge so ein, dass sie mit einem Widerstand von 15 kΩ angesteuert werden, und mit der letzten Zeile stellen wir

#### Listing 1. Pin-Einstellung.

```
WRPIN(52, %0000_0000_000_010_010_00_00000)
WRPIN(53, %0000_0000_000_010_010_00_00000)
WRPIN(54, %0000_0000_000_010_010_00_00000)
WRPIN(55, %0000_0000_000_010_010_00_00000)
```

#### Listing 2. Byte-Deklaration.

VAR BYTE PinA, PinB, PinC, PinD

unsere vier Eingangspins so ein, dass sie auf Low geschaltet werden, da wir sie tatsächlich als Ausgang verwenden.

Nach der Initialisierung wird die Funktion read\_switch() wiederholt aufgerufen, so dass wir einen genaueren Blick auf den in dieser Funktion ablaufenden Code werfen sollten. Wie in **Listing 5** zu sehen, verwendet der Code nach dem Einlesen der Pin-Zustände in die Variablen a...d einige *IF*- und *IF-ELSE*-Konstrukte. Zunächst muss festgestellt werden, ob sich ein Pin geändert hat, indem der gespeicherte alte Zustand mit dem gerade eingelesenen verglichen wird. Wenn sich diese Werte unterscheiden, können wir davon ausgehen, dass sich der Zustand am Eingangspin geändert hat, und feststellen, ob dies darauf zurückzuführen ist, dass der entsprechende Schalter gedrückt oder losgelassen wurde. Je nach aktuellem Zustand des Pins verwenden wir die Funktion prints(), um "Switch pressed" oder "Switch released" auszugeben. In Listing 5 ist auch zu sehen, wie unsere **read\_switch()**-Funktion aussieht und funktioniert, mit einfachen

# Listing 3. Initialisierungsfunktionen hinzugefügt. PUB main() LED\_init() Input\_init() pinwrite(56, 1 ) serial\_start() repeat read\_switch()
#### Listing 4. Eingangsinitialisierung.

```
PUB Input_init( )
' A and B as input using A with 15k drive
 strength for pull-up and pull-down as output no
' smartpin function
  WRPIN(52, %0000_0000_000_010_010_000_0000)
  WRPIN(53, %0000_0000_000_010_010_00_00000)
  WRPIN(54, %0000_0000_000_000_010_010_00_00000)
  WRPIN(55, %0000_0000_000_010_010_00_00000)
  PINWRITE(55..52,0)
  'Drive pin low with 15k resistance
```

#### Listing 5. Die Funktion read\_switch().

```
PUB read_switch() | a,b,c,d
'We will read the pin, reflect the LED state
'We send a change in state with the UART
 a := pinread(52)
 b := pinread(53)
 c := pinread(54)
 d := pinread(55)
 IF(PinA <> a )
    PinA := a
     prints(string("A:"))
    IF( a <> 0 )
      pinwrite(49,1)
       prints(@PinPressed)
     ELSE
      pinwrite(49,0)
      prints(@PinReleased)
 IF(PinB <> b )
    PinB := b
    prints(string("B:"))
    IF( b <> 0 )
       pinwrite(48,1)
       prints(@PinPressed)
     ELSE
       pinwrite(48,0)
       prints(@PinReleased)
 IF(PinC<> C )
    PinC := c
     prints(string("C:"))
     IF( c <> 0 )
      pinwrite(50,1)
     prints(@PinPressed)
     ELSE
      pinwrite(50,0)
      prints(@PinReleased)
  IF(PinD <> d )
    PinD := d
     prints(string("D:"))
    IF(d \leftrightarrow 0)
      pinwrite(51,1)
      prints(@PinPressed)
     ELSE
       pinwrite(51,0)
       prints(@PinReleased)
```



Bild 9. Propeller 2 mit angeschlossenen Tasten.

IF-THEN-ELSE-Pfaden für jeden verwendeten Schalter. Man kann dort auch erkennen, dass der Zustand der LEDs entsprechend dem neu erkannten Zustand geändert wird.

#### Zum Schluss: erste Eindrücke

Damit haben wir erst einmal genug mit den Pins experimentiert. Ich hoffe, Sie haben einen kleinen Einblick in die Funktionsweise gewonnen. Und dies ist auch das vorläufige Ende der Artikelreihe. Doch da sich der Propeller 2 stets in Reichweite meines Labortisches befindet, wird es garantiert nicht das letzte Mal sein, dass wir darüber schreiben oder ein kurzes Video drehen. Da sich die Software und die IDE weiterentwickeln, sollten wir ihr etwas Zeit geben, um erwachsen zu werden. In der Zwischenzeit können Sie auf den in dieser Artikelreihe entwickelten Code in unserem GitHub-Repository [1] zugreifen.

Was bleibt nun abschließend zu sagen? Nachdem ich mit dem Propeller 2 etwas gearbeitet habe, kann ich sagen, dass es sich um eine leistungsstarke MCU handelt, die definitiv mehr Aufmerksamkeit für ihre Details benötigt als ein Raspberry Pi Pico. Die Verwendung von SPIN2 und Assembler ist etwas Besonderes und macht das Recycling von bestehendem C-Code unmöglich. Persönlich denke ich, dass es nicht die MCU meiner Wahl für die Mainstream-Projekte sein wird, aber es gibt definitiv einige Nischen, in denen sich dieser Chip als nützlich erweisen kann.

Und was ist mit den eingangs erwähnten SPI, I<sup>2</sup>C und HDMI? Da es einige Verbesserungen in den Toolchains gibt, wäre es interessant, dies mit verbesserten Tools auszuprobieren. Wenn Sie nicht auf das nächste Kapitel über den Propeller 2 warten wollen, schreiben Sie uns eine E-Mail oder nutzen Sie einen Ihrer bevorzugten sozialen Kanäle, um uns eine Nachricht zu senden. - |◀

200479-E-03

#### **Ein Beitrag von**

Entwurf und Text: Mathias Claußen Redaktion: Jens Nickel, C. J. Abate Übersetzung: Rolf Gerstendorf Layout: Giel Dols

#### Haben Sie 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 oder kontaktieren Sie Elektor unter redaktion@elektor.de.





Teil 1: Hardware und Programmierung

Von Josef Bernhardt und Martin Mohr

Das Modbus®-Protokoll ist ein in der Industrie weit verbreitetes Verfahren, mit dem Anlagen und Steuerungen untereinander kommunizieren. Üblicherweise geschieht das über die gute alte RS485-Schnittstelle und traditionelle



Modbus-Platine mit NodeMCU-Modul. Modbus® ist eine eingetragene Marke von Schneider Electric, lizenziert an die Modbus Organization, Inc.

Kupferadern. In diesem Artikel stellen wir ein Modul vor, mit dem man das Modbus-Protokoll drahtlos über WLAN verwenden kann. Kern des Moduls ist ein NodeMCU-Board mit dem Mikrocontroller ESP8266. Eine zusätzliche Modbus-Basis-Platine sorgt vor allem dafür, dass man mit der in der Industrie üblichen Spannung von 24 V arbeiten kann. Zur anschaulichen Demonstration bauen wir eine einfache Garagentorsteuerung.

Das NodeMCU-Modul und die Arduino-IDE dürften den meisten Elektor-Lesern ja wohlbekannt sein. Wenn Sie dazu gehören, können Sie diesen Abschnitt auslassen und direkt zur Beschreibung der Modbus-TCP-Platine springen. Für alle anderen nun ein Crashkurs! Der Kern dieses Projektes ist ein NodeMCU- Modul, das auch bei Elektor erhältlich ist. Auf dem Modul ist ein ESP8266-Mikrocontroller verbaut, der kaum größer ist als eine Briefmarke und über eine WLAN-Schnittstelle verfügt. Die Rechenleistung ist auch nicht zu verachten. In **Tabelle 1** sind die wichtigsten technischen Daten des ESP8266-Mikrocontrollers zusammengefasst. Das NodeMCU-Board erzeugt die Betriebsspannung für den ESP8266 und kümmert sich um die Ansteuerung der Programmierschnittstelle des ESP8266. Eine Übersicht der Anschlüsse des NodeMCU-Boards, wie sie in unserer Modbus-Schaltung verwendet werden, sehen Sie in **Bild 1**. Zur Programmierung des NodeMCU-Boards eignet sich hervorragend die Arduino-IDE, die frei von der Arduino-Homepage [2] passend für das Betriebssystem Ihres Rechners heruntergeladen und gemäß der Anweisungen auf der Webseite installiert werden kann. Nach dem ersten Start präsentiert sich die IDE wie in Bild 2 zu sehen: Im Bereich für den Programmcode sind schon zwei Funktionen vordefiniert, die setup()-Funktion, die nur einmal beim Start des Programms ausgeführt wird und die beispielsweise der Initialisierung der Mikrocontroller-Schnittstellen dient, und unterhalb davon die loop()-Funktion, die normalerweise den Sourcecode Ihres Programms beherbergt.

Die loop()-Funktion wird aufgerufen, sobald die setup()-Funktion beendet ist. Ist die loop()-Funktion an ihr Ende gelangt, startet sie immer wieder neu. Zwischen dem Beenden und dem Neustarten der loop()-Funktion verwaltet der ESP8266 die WLAN-Schnittstelle. Daher müssen Sie darauf achten, dass sich in der loop()-Funktion keine Endlosschleife befindet, denn dies würde den ESP8266 unweigerlich zum Absturz bringen. Der Code in der loop()-Funktion muss vielmehr so beschaffen sein, dass er zyklisch abgearbeitet werden kann.

Ein Großteil aller schwer nachvollziehbaren Abstürze des ESP8266 haben ihre Ursache darin, dass der ESP8266 nicht die notwendige CPU-Zeit zur Verwaltung der WLAN-Schnittstelle erhält. Wenn Ihr Programmcode potentiell länger läuft, zum Beispiel in großen Schleifen, können Sie dem ESP8266 mit der yield()oder der delay()-Funktion die Zeit zur WLAN-Verwaltung geben.

Das NodeMCU-Modul wird (noch nicht auf die Modbus-Platine gesteckt, sondern direkt) an den USB Ihres Computers angeschlossen, aber eines muss noch geschehen, bevor Sie loslegen können. Der ESP8266 wird standardmäßig nicht von der Arduino-IDE unterstützt, sondern muss nachgerüstet werden. Dazu gibt man unter Datei → Voreinstellungen im Feld Zusätzliche Bordverwalter-URLs die URL: http://arduino.esp8266.com/stable/package\_ esp8266com\_index.json an, öffnet nach dem OK den Boardverwalter unter Werkzeuge →Board: "..." → Boardverwalter..., sucht und findet ESP8266 und installiert die Boards der ESP8266 Community.

Nach der Installation wird unter Werkzeuge → Board das Board NodeMCU 1.0 (ESP-E12 *Module*) und unter *Werkzeuge* → *Port* der Port eingestellt, an dem das NodeMCU-Board angeschlossen ist (Linux: /dev/ttyUSBx, Windows COMx).



Bild 1. Anschlussbelegung des NodeMCU-Moduls für das Modbus-Projekt.

#### Tabelle 1. Technische Daten des ESP8266.

32-Bit-RISC-CPU	Tensilica Xtensa LX106 mit 80 MHz
Programmspeicher	64 KB
Datenspeicher	96 KB
Externer Flashspeicher (optional)	512 KB bis 4 MB
GPIO	10 Pins, mögliche Schnittstellen: SPI, I²C, I²S, UART. Es kann DMA und IRQ verwendet werden.
WLAN	- IEEE 802.11 b/g/n- WEP oder WPA/WPA2- Authentifizierung- Wi-Fi Direct (P2P), Soft-AP- Integrierter TCP/IP-Protokoll-Stack
ADC	1x 10-Bit-Wandler
Antenne	integriert
Ausgangsleistung	+19,5 dBm im 802.11b-Modus
Größe	17,2 x 12,3 mm <sup>2</sup>
Standby-Strom	< 1,0 mA

Jetzt können wir mit einem ersten kleinen Test des NodeMCU-Moduls beginnen. Öffnen Sie dazu bitte das Blink-Beispielprogramm  $(Datei \rightarrow Beispiele \rightarrow ESP8266 \rightarrow Blink).$ Das Programm in Listing 1 lässt die auf dem ESP8266-Board verbaute LED blinken. Der Code, der die LED ein- und ausschaltet, befindet sich in der loop()-Funktion. In der setup()-Funktion wird der GPIO-Pin der LED als Ausgang gesetzt.

Um das Programm auf das NodeMCU-Board zu laden, klicken Sie nun auf das Hochladen-Icon (Pfeil nach rechts). Nach einigen

#### Listing 1. Blink für den ESP8266.

```
void setup() {
 pinMode(LED_BUILTIN, OUTPUT);
3
void loop() {
 digitalWrite(LED_BUILTIN, LOW);
 delay(1000);
 digitalWrite(LED_BUILTIN, HIGH);
  delay(2000);
```





Sekunden ist das Programm übertragen und die LED beginnt zu blinken. Mit diesem Test ist die Installation der IDE abgeschlossen.

#### **Die Modbus-TCP-Platine**

Wie im Titelbild zu sehen ist, befindet sich das NodeMCU-Modul auf der Modbus-TCP-Platine, auf der es recht "industriell" zugeht.

Bild 2. Arduino-IDE nach dem Start.

Alle Ein- und Ausgänge sind mit steckbaren Schraubklemmen versehen. Das erleichtert die Montage und die Schraubverbinder können leicht getauscht werden, wenn mal einer kaputtgeht. Hauptsächliche Aufgabe der Modbus-TCP-Platine ist es, die 3,3 V der Mikrocontroller-Ein- und Ausgänge in industriekompatible 24 V umzuwandeln. Das Schalt**bild 3** zeigt, dass die vier Eingänge durch PC814-Optokoppler galvanisch vom Controller getrennt sind. Die Eingänge sind nicht einzeln am Controllerboard angeschlos-





Bild 4. Interner Aufbau des BSP76 (Quelle: Infineon-Datenblatt).



Bild 5. Ein hölzernes Garagentor aus China.

sen, sondern über einen Multiplexer 74HC151. Mit diesem kleinen Trick spart man einen GPIO-Anschluss, an den man einen DHT11-Sensor zur Messung von Temperatur und Luftfeuchte in die Schaltung aufzunehmen kann. Damit ist es möglich, die Modbus-Platine für einfache Klimaregelungen zu verwenden.

In den Ausgangsstufen sind BSP76 von HITFET verbaut (Bild 4). Dabei handelt es sich nicht um simple Power-MOSFETs, sondern um ICs, die den elektronischen Leistungsschalter mit verschiedenen Schutzmechanismen wie Überspannungsschutz, Strombegrenzung und Übertemperaturschutz ausstatten. So kann man ohne Bedenken ohmsche, induktive und kapazitive Lasten schalten.

Neben den digitalen Ein- und Ausgängen verfügt die Modbus-TCP-Platine über einen analogen Eingang und einen analogen Ausgang. Beide Analoganschlüsse arbeiten mit dem in der Industrie üblichen Spannungsbereich von 0...10 V. Beim analogen Eingang reduziert der Spannungsteiler R14/R17 die Eingangsspannung auf maximal 3,3 V. Der Eingang ist nicht galvanisch getrennt, aber C7, D10 und D11 sorgen für einen gewissen Schutz gegen Spikes und Überspannungen. Die Reihenwiderstände R14 und R19 begrenzen den Eingangsstrom, so dass die Dioden im Falle eines Falles nicht in Rauch aufgehen können.

Beim analogen Ausgang hebt IC3, ein Universal-Operationsverstärker LM7301 (Rail-to-Rail an Ein- und Ausgängen) die Ausgangsspannung des DAC des NodeMCU-Moduls an. Die Verstärkung ist durch R15 und R16 auf den Faktor 3 eingestellt. Der Opamp wird hier mit 24 V versorgt.

Besonderes Augenmerk sollten wir auf den DC/DC-Wandler von Recom richten. Er wandelt die 24-V-Versorgungsspannung in die 5 V, die für die Spannungsversorgung des NodeMCU-Moduls benötigt werden. Der DC/DC-Wandler weist einen Wirkungsgrad von weit über 90% auf und kann sehr flexibel eingesetzt werden. Es ist zum Beispiel auch möglich, mit dem Wandler eine negative Betriebsspannung für Operationsverstärker zu erzeugen. Ein Blick in das Datenblatt [4] des Wandlers lohnt sich auf jeden Fall. Direkt am 24-V-Eingang befindet sich eine 1N4007-Diode als Verpolungsschutz für die Schaltung. Die Gerber-Dateien für die Platinenproduktion können von der Projektseite [8] heruntergeladen werden. Die Platine ist im unbestückten Zustand oder auch betriebsfertig bestückt beim Autor erhältlich [7].

#### **Holzmodell zur Demonstration**

Um zu zeigen, wie das Modbus-Modul mit in der Industrie üblichen Aufgaben fertig wird, habe ich ein Garagentor-Modell frisch aus dem chinesischen Supermarkt [5] verwendet (Bild 5). Das Tor wird über zwei Drucktaster geöffnet und geschlossen. Die Endpositionen des Tores werden über zwei kapazitive Näherungsschalter erkannt. Der Motor, der das Tor öffnet und schließt, wird mit Hilfe einer H-Brücke angesteuert, die aus zwei Leistungsrelais aufgebaut ist.

In Listing 2 ist das Programm zur Ansteuerung zu sehen. Das Programm definiert zunächst Klartext-Namen für die Ein- und Ausgänge der Schaltung. In der setup()-Funktion werden die einzelnen I/Os als Ein- und Ausgänge initialisiert. Danach folgen die Funktionen input() und output(), die uns einen etwas einfacheren Zugriff auf die I/Os ermöglichen. Das eigentliche Programm befindet sich in der

#### **WEBLINKS**

- NodeMCU-Board: www.elektor.de/ [1] nodemcu-microcontroller-board-with-esp8266-and-lua
- Arduino-IDE Download: www.arduino.cc/en/software [2]
- Datenblatt BSP76: https://bit.ly/3r3GP99 [3]
- Datenblatt DC/DC-Wandler: https://recom-power.com/pdf/Innoline/R-78-0.5.pdf [4]
- Garagentor-Holzmodell: https://de.aliexpress.com/item/33008243573.html [5]
- [6] Garagentormodell auf Youtube: https://youtu.be/VHIBQswdA0E
- Josef Bernhardt: www.bernhardt.de [7]
- [8] Projektseite: www.elektormagazine.de/200507-02



Listing 2. Garagentorsteueru	ng.	
#define A1 16	<pre>pinMode(A2, OUTPUT);</pre>	
#define A2 5	<pre>pinMode(A3, OUTPUT);</pre>	
#define A3 19	<pre>pinMode(A4, OUTPUT);</pre>	<pre>void output(int i, int v){</pre>
#define A4 0	<pre>pinMode(MUXA, OUTPUT);</pre>	<pre>digitalWrite(i,v);</pre>
#define MUXA 12	<pre>pinMode(MUXB, OUTPUT);</pre>	}
#define MUXB 13	<pre>pinMode(MUXY, INPUT);</pre>	
#define MUXY 14	}	
#define E1 0		<pre>void loop() {</pre>
#define E2 1		<pre>if(input(E4)&amp; !input(E1)){output(A1,HIGH);}</pre>
#define E3 2	<pre>bool input(int i){</pre>	<pre>if(input(E3)&amp; !input(E2)){output(A2,HIGH);}</pre>
#define E4 3	<pre>digitalWrite(MUXA,(i&amp;1));</pre>	<pre>if(input(E1)){output(A1,LOW);}</pre>
	<pre>digitalWrite(MUXB,(i&amp;2)&gt;&gt;1);</pre>	<pre>if(input(E2)){output(A2,LOW);}</pre>
	<pre>delay(1);</pre>	}
<pre>void setup() {</pre>	<pre>return digitalRead(MUXY);</pre>	
<pre>pinMode(A1, OUTPUT);</pre>	}	

loop()-Funktion. Hier werden die Ausgänge zur Ansteuerung des Motors abhängig von den Tastern und Endschaltern an- oder ausgeschaltet. Das Programm in der loop()-Funktion kann zyklisch ausgeführt werden. Falls Sie die Garagentürsteuerung in Aktion sehen wollen, schauen Sie doch einfach einmal bei YouTube [6] vorbei.

#### **Ausblick**

Ihnen dürfte trotz der Freude über die funktionierende Garagentor-Steuerung nicht entgangen sein, dass unser WLAN-Modul noch gar nicht zum Zuge gekommen ist. Dies wird der Inhalt des zweiten Teils des Artikels sein! Wir beschäftigen uns dann auch mit dem Modbus-Protokoll und welche Software man einsetzen kann, um über den Bus zu kommunizieren. Weiterhin zeigen wir, wie das Modbus-Modul eingerichtet werden kann.

200507-02



Idee und Ausführung: Josef Bernhardt, Martin Mohr

Redaktion: **Rolf Gerstendorf** Layout: **Giel Dols** 

#### Sie haben Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare? Wenden Sie sich bitte an den Autor unter josef@bernhardt.de oder an die Redaktion unter editor@elektor.com.

#### **Die Autoren**

Josef Bernhardt hat sich schon früh mit Elektronik beschäftigt. Mit zwölf Jahren baute er seinen ersten Detektorempfänger, weitere Schaltungen folgten. Die ersten Programmiererfahrungen sammelte er in den 80er-Jahren mit dem Commodore VC20. Auch Assemblerprogrammierung auf dem 8088 Prozessor ist ihm nicht fremd. Er kann auf über 30 Jahre Elektronikentwicklung an der Uni Regensburg zurückblicken. Dort war er im Bereich Elektronik- und Softwareentwicklung tätig. Mit der eigenen SMD-Fertigung setzt er auch Elektronikprojekte von Kunden um.

Martin Mohr erblickte im Zeitalter der Magnet-Ringkernspeicher und Hebdrehwähler das Licht der Welt und hat somit die komplette Entwicklung der modernen Computertechnik miterleben dürfen. Die Vorliebe für alles, was blinkt, hat sich schon in seiner frühen Jugend entwickelt und wurde durch eine Ausbildung zum Elektroniker noch verstärkt. Nach dem Informatikstudium lag sein Beschäftigungsfeld überwiegend in der Entwicklung von Java-Applikationen. Mit dem Raspberry Pi ist aber die alte Liebe zur Elektronik wiedererwacht.



PASSENDE PRODUKTE

NodeMCU-Modul

www.elektor.de/nodemcu-microcontroller-board-with-esp8266-and-lua

embeddedworld2022

Exhibition&Conference



## INTELLIGENT. CONNECTED.EMBEDDED.

JOIN THE GLOBAL PLATFORM OF THE EMBEDDED COMMUNITY 15 – 17.3.2022

You simply have to be there! embedded-world.com

Media partners

#### Markt&Technik

Elektronik

Smarter World





Elektronik automotive

•medical-derign

elektroniknet.de







## Zutritt für Unbefugte verboten

Wie der Junior-Computer wieder zum Leben erweckt wurde

Von Laurent Francoise (Frankreich) und Eric Bogers (Elektor)

Im Jahr 1980 ging ein Schock durch die Elektronikwelt - im Mai jenes Jahres stellte Elektor den Junior-Computer [1] vor und betrat damit ein für viele Elektronikbastler unbekanntes Terrain. Das Computerzeitalter hatte gerade erst begonnen, und für viele war so ein "Ding" unheimlich und teuer. Kaum jemand wusste damit etwas anzufangen. Mit dem Junior-Computer änderte sich das - er war nicht nur relativ erschwinglich (damals etwa 500 DM), sondern man konnte (musste) ihn auch selbst zusammenbauen.



Bild 1. Der Junior-Computer erblickt das Licht der Welt.



Das Projekt Junior-Computer (**Bild 1**) war so umfangreich, dass es nicht möglich war, alles in der Zeitschrift zu veröffentlichen. Der größte Teil der Beschreibung erschien in einer Reihe von Elektor-Büchern. Diese sind schon lange nicht mehr erhältlich, aber wer im Internet sucht, wird feststellen, dass mehrere Enthusiasten zumindest Teile davon digitalisiert und Interessierten zur Verfügung gestellt haben (danke!). Auch die unübertroffene Kolumne "Retronik" hat 2005 auf "den JC" aufmerksam gemacht [2].

In Frankreich hat es Elektor-Leser Laurent Francoise, dessen liebstes Hobby die Restaurierung alter Computer ist, auf sich genommen, dem Junior-Computer wieder Leben einzuhauchen. Erteilen wir Laurent das Wort:

"Ich bin in den achtziger Jahren aufgewachsen und habe von meinem Vater (der Funkamateur war) die Liebe zur Elektronik und zu Computern geerbt. Die Sammlung der Elektor-Zeitschriften (von der allerersten Ausgabe an!) nahm einen wichtigen Platz in meinem Leben ein; unzählige Male habe ich sie durchgeblättert, obwohl ich damals nicht alles verstanden habe. Viele der Projekte, die mein Vater damals gebaut hat (Elekterminal, BASIC-Mikrocomputer, ElektorScope), sind noch immer in meinem Besitz.

Vor nicht allzu langer Zeit stieß ich wieder auf das Junior-Computer-Buch, und ich bekam Lust, den Junior zu bauen. Es wäre fantastisch gewesen, wenn ich damals die Platinen dafür bestellt hätte (im berühmten Elektor-Blau) – doch die sind jetzt nirgends mehr zu finden."



Für Leser, die mit dem Junior nicht vertraut sind: Es handelte sich um einen Einplatinencomputer mit einem 6502 als CPU, nicht weniger als 1 kB ROM und 1 kB RAM (plus 128 Byte RAM im I/O-Chip Typ 6532), einer kleinen Tastatur (mit den berühmten Digitastern), einem 6-stelligen 7-Segment-Display und einem Anschluss an der Seite, der Zugang zum Systembus bot - praktisch für spätere Erweiterungen (und die gab es reichlich). **Bild 2** zeigt das Blockdiagramm (sorry für die schlechte Scan-Qualität).

"Ich musste also die Platinen selbst herstellen. Damals war es üblich, das schwarz-weiß gedruckte Layout in der Zeitschrift auf Transparentfilm zu kopieren und dann diesen Film zu verwenden, um eine mit Fotolack beschichtete Platine zu belichten, zu entwickeln und zu ätzen. Selbst doppelseitige Platinen waren kein unüberwindbares Problem – nur: Die Platine für den Junior hatte durchkontaktierte Löcher, und das macht eine Heimfertigung unmöglich. Theoretisch könnte man versuchen, mit dünnen Kupferlitzen aus feinem Litzendraht die durchkontaktierten Löcher selbst zu machen, doch das bedeutet, um Probleme zu betteln. Das kommt für mich nicht in Frage.

Ich musste also die Platinen nachbauen und anfertigen lassen, was heutzutage überhaupt kein Problem mehr ist. Dabei wollte ich das



Bild 4. Die rekonstruierten Platinen des Junior Computers sind von den Originalen fast nicht zu unterscheiden.



Bild 6: Der rekonstruierte Junior-Computer in seiner ganzen Pracht. Alles funktioniert!

Original so genau wie möglich nachbilden. Ich habe zunächst die berühmten achteckigen Elektor-Pads in KiCAD erstellt, ebenso wie die charakteristischen Elektor-Footprints, die benötigt wurden. Danach importierte ich einen Scan des Original-Layouts als Layer in KiCAD. Die ,Blaupause' stellte sicher, dass alle Bauteile so weit wie möglich an ihren ursprünglichen Position waren (Bild 3). Es erwies sich als überraschend einfach, dies in KiCAD zu bewerkstelligen. Das Routing der Leiterbahnen habe ich von Hand gemacht. Das Ergebnis all dieser Bemühungen ist in Bild 4 zu bewundern - einschließlich des Elektor-Blaus. Während ich daran arbeitete, stellte sich mir eine wichtige Frage: Wollte ich eine exakte Replik des Original-Juniors machen, oder waren ein paar Anpassungen akzeptabel? Ich habe lange darüber nachgedacht und mich schließlich entschlossen, einige der späteren Verbesserungen und Korrekturen, die in Elektor veröffentlicht wurden, in meinen Nachbau aufzunehmen. Außerdem habe ich das Original-EPROM 2708 durch ein 2716 ersetzt. Das brachte zwei Vorteile: erstens ist der 2716 bei eBay leichter zu bekommen als der 2708, und außerdem benötigt der 2716 im Gegensatz zum 2708 nur eine einzige Versorgungsspannung von +5 V - der 2708 benötigt zusätzlich -5 V und +12 V. Das bedeutete natürlich, dass ich auch den ursprünglichen Schaltplan in KiCAD neu zeichnen musste, um die Änderungen zu berücksichtigen. Dies war jedoch nur eine Formalität. Ein Teil des neuen Schaltplans ist in **Bild 5** zu sehen.

Die schwierigste Arbeit habe ich mir für den Schluss aufgehoben - die Beschriftung der Digitaster. Es war ein unglaubliches Glück, dass ich die originalen Mecanorma-Abreibesymbole zur Verfügung hatte; mit diesen gelang es mir, eine dem Original identische Tastatur zu schaffen. Das Endergebnis kann in **Bild 6** bewundert werden.

Der Zusammenbau war recht einfach; der Junior funktionierte fast sofort – der 6532, den ich einbaute, erwies sich als fehlerhaft, aber zum Glück hatte ich zwei bestellt. Danach konnte ich den Junior ausgiebig testen, einschließlich eines kleinen Programms, das "Junior' auf die 7-Segment-Anzeigen schreibt. Ich bin sehr froh, dass dieses Projekt so gut gelungen ist! Und weil so viele Erweiterungen für den Junior-Computer erschienen sind (Videokarten und sogar ein Floppy-Disk-Controller, um nur zwei zu nennen), denke ich, dass ich noch eine ganze Weile viel Freude an dem Junior-Computer haben werde!"

210328-03

#### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Anmerkungen zu diesem Artikel? Senden Sie eine E-Mail an den Autor oder die Redaktion von Elektor über redaktion@elektor.de.

#### Ein Beitrag von

Text und Illustrationen: Laurent Francoise Redaktion: Eric Bogers Übersetzung: Sophia Gerstendorf Layout: Harmen Heida

#### WEBLINKS

[1] "Junior-Computer", Elektor, Mai 1980: https://www.elektormagazine.de/magazine/elektor-198005

[2] "Elektor Junior.Computer" (Retronik), Elektor, Januar 2005: https://www.elektormagazine.de/magazine/elektor-200501/2044



Bild 5. Ein Teil des leicht veränderten Schaltplans.

## Bauen Sie Ihren eigenen Hochpräzisionskalibrator

#### Von Vincent Gautier (Frankreich)

Bei der Reparatur oder beim Abgleich elektronischer Schaltungen kann eine hochpräzise Spannungs- oder Stromreferenz erforderlich sein. Der Kalibrator, den wir in diesem Artikel vorstellen, erzeugt genaue Spannungen von -10 V bis +10 V in Schritten von 20  $\mu$ V und genaue Ströme von 0 bis 40 mA in Schritten von 100 nA zum Testen Ihrer Entwürfe oder als Kalibrierungsquelle für Geräte wie Multimeter oder Oszilloskope.

#### **Eigenschaften und Spezifikationen**

- Ausgangsspannung: 0 bis +/- 10 V mit 20 μV Auflösung, max. 5 mA
- > Strom: 0 bis 40 mA, Auflösung 100 nA, Leerlaufspannung 12 V
- > Fehler: 10 ppm (parts per million) bei 21°C
- > Einfach per Software zu kalibrieren
- > Stromversorgung über USB (Batterie optional)
- > 3,5" Farb-Touchscreen



Dieser Artikel basiert auf dem Projekt von Vincent Gautier auf Elektor-Labs [1]. Dort finden Sie detaillierte Informationen zu seinem Kalibrator einschließlich der gesamten Software, der Platinenlayouts, der Stücklisten und der 3D-Druckerdateien, die Sie allesamt herunterladen können. Ein Video, in dem dieses Projekt vorgestellt wird, ist unter [2] zu finden.

Ich habe ein kaputtes Oszilloskop auf eBay gekauft. Um seine Funktionen nach der Reparatur voll ausnutzen zu können, musste es mit präzisen Spannungen und Strömen kalibriert werden. Normalerweise liegen die benötigten Kalibrierwerte im Bereich von 1,000 V oder 10,000 mA. Für dieses Oszilloskop werden jedoch gemäß den vom Hersteller angegebenen Kalibrierverfahren spezifische Spannungs- und Stromreferenzen von +1,7694 V, -1,7694 V und 20,253 mA benötigt.

Es gibt viele Kalibrierlösungen auf dem Markt, aber diese Geräte liegen entweder weit außerhalb meines Budgets oder sie haben nicht die erforderliche Auflösung und Genauigkeit. Die Alternative für mich war also der Bau eines eigenen Kalibrators.

#### **Die Hardware**

- Der Entwurf besteht aus sechs Funktionsblöcken:
- Stromversorgung
- > Spannungsreferenz
- DAC um den AD5791
- Stromgenerator
- > Mensch-Maschine-Schnittstelle (HMI) mit ihrem Mikrocontroller
- > Heizelement zur Stabilisierung der Referenzspannung

Die Schaltung der fünf ersten Blöcke ist in **Bild 1** zu sehen. Auf die separate Heizung wird später eingegangen.

Das HMI-Modul stellt die Benutzerschnittstelle dar und stellt die Ausgangsspannung des DAC ein. Diese hochgenaue Spannung wird vom Stromgenerator in einen ebenso hochgenauen Ausgangsstrom an den Klemmen J6 und J7 umgewandelt, wenn das HMI dies über FET Q1 und Relais K1 so will. Die Genauigkeit wird größtenteils durch die Spannungsreferenz bestimmt, die eine wichtige Voraussetzung für diesen Entwurf ist.

#### **Die Stromversorgung**

Der Kalibrator wird über ein USB-Netzteil mit mindestens 500 mA versorgt. Die Betriebsspannung von 5 V wird in drei separate Zweige aufgeteilt, von denen einer direkt das HMI mit dem Mikrocontroller



versorgt. Im zweiten Zweig sorgt U1, ein 3,3-V-LDO, für die 3,3-V-Betriebsspannung des DAC U5. Der dritte Zweig wird zu PS1 geführt, einem galvanisch getrennten, symmetrischen DC/DC-Wandler mit einer Ausgangsspannung von  $\pm$ 15 V. Diese Ausgangsspannungen werden gefiltert und versorgen die beiden Linearregler U2 und U3. Bei beiden Reglern handelt es sich um Low-drop-Typen mit geringem Rauschen. Ihre Ausgangsspannungen von  $\pm$ 13,5 V respektive -13,5 V werden für die analogen Ausgangsstufen der DAC-Schaltung verwendet.

#### Die Spannungsreferenz

Dies ist das wichtigste und zugleich kritischste Element in diesem Projekt. Wenn wir eine Referenzspannung von 10,00000 V mit einem Fehler von ±10 µV zur Verfügung hätten, würden wir uns wünschen, dass es überall, für immer und unter allen Umständen bei dieser Spannung und Präzision bleibt. In der realen Welt ist das leider nur Wunschdenken. Das Thema Entwurf genauer und stabiler Spannungsreferenzen ist eine Wissenschaft für sich. Im Internet gibt es Dutzende von Artikeln und Fachleuten, die sich mit diesem Thema befassen, um die Grenzen von Merkmalen wie Rauschen, kurz- und langfristige Stabilität in ppm/°C, Hysterese, thermoelektrische EMK, piezoelektrische Eigenschaften und so weiter auszuloten.[4][5]. Einige Autoren haben auch Spannungsreferenzen mit erstaunlich schönen Platinen mit Widerständen mit extrem niedrigem Temperaturkoeffizienten in ölgefüllten Metallgehäusen hergestellt.



$$V_{out} = \frac{\left(+V_{ref} - \left(-V_{ref}\right)\right)}{2^{20} - 1} \times code - V_{ref}$$

Oder ungefähr:

$$V_{out} = 20 \mu V \times code - V_{ref}$$

Suchen Sie mit Ihrer Lieblingssuchmaschine einmal nach den Werten (und den Preisen) für den Rolls Royce unter den Spannungsreferenzen, den LTZ1000 mit einer Drift von nur 0,05 ppm/°C [6]. Nachdem ich drei etwas günstigere Spannungsreferenzen (LTC6655, ADR445, MAX6350) getestet hatte, entschied ich mich für die MAX6350ESA von Maxim, die sich als die zeitstabilste mit 5-V-Ausgang erwies. Mit den beiden Operationsverstärkern U10A/B im AD8676 (dualer rauscharmer Op-Amp mit geringer Offset-Drift) werden +10,48 V und -10,48 V als Referenzen VREF+ beziehungsweise VREF- erzeugt. Die Widerstände R16...R20 um den AD8676 sind Widerstände mit geringer Temperaturdrift (besser als 2 ppm/°C).

#### Der DAC AD5791

Bild 2. Screendump der Workshop-4-IDE.

Der von mir eingesetzte Digital-Analog-Wandler ist ein AD5791, der wegen seiner Auflösung von 20 Bit und der Linearität von 1 ppm ausgewählt wurde. Eine solche Linearität vereinfacht die Kalibrierung: Nur die Punkte 0 V und 10 V müssen angepasst werden. Die Beschaltung des AD5791 entspricht der des *Analog Devices Evaluation Boards* für diesen DAC [7]. In meinem Entwurf wird der DAC mit Referenzspannungen von +10,48 V und -10,48 V, was Schritte von etwa 20 µV ergibt. Die Ausgangsspannung kann berechnet werden mit: Die Ausgangsspannung des Kalibrators wird von der Software auf einen Wert zwischen -10 V und +10 V begrenzt. Dies bietet eine große Möglichkeit, den Offsetfehler bei 0 V zu kompensieren.

#### **Die Präzisionsstromquelle**

Die vier Hauptkomponenten der Präzisionsstromquelle sind U7 (Instrumentenverstärker AD8276), Q1 (NPN-Transistor PZT1222), U8 (Präzisionsverstärker AD8677) und R10 (100  $\Omega$ ). Q1 liefert den Strom an die Last. Dieser Strom fließt durch den Shunt-Widerstand R10 zum Relais und dann zur zur testenden Last, um schließlich zur Masse zurückzukehren. Die von U8 und U7 gemessene Spannung über R10 wird mit der vom DAC AD5791 gelieferten Referenzspannung verglichen. Der Strom der Ausgangsquelle ist gleich V<sub>DAC</sub> / R10 oder V<sub>DAC</sub> / 100. Der Strom wird von der Software auf 40 mA begrenzt, um eine Drift der Spannungsmessung aufgrund der thermischen Erwärmung von R10 und Q1 zu vermeiden. Der Temperaturkoeffizient des Widerstands R10 ist kritisch und muss so niedrig wie 0,2 ppm/°C sein. Aus diesem Grund ist auf der Platine Q1 auch so weit wie möglich von R10 und natürlich von den Referenzspannungen entfernt.



86 November/Dezember 2021 www.elektormagazine.de

#### Das HMI und der Mikrocontroller

Bei dieser Komponente handelt es sich um ein gen4-uL-CD35-DT-Modul von 4D Systems. Das Modul vereint einen resistiven 3,5-Zoll-Farb-Touchscreen, einen Mikrocontroller und einen micro-SD-Kartenhalter. Die IDE 4D Workshop 4 [3] ist kostenlos und enthält die Touch-Button-Funktionsbibliotheken für dieses Projekt (Bild 2). Der Code selbst ist einer vereinfachten Form von C geschrieben. Die grafische Benutzeroberfläche (GUI) ist wie folgt aufgebaut: Touch-Tasten und Ziffern sind Grafiken, die auf der Micro-SD-Karte gespeichert sind. Jede Grafik ist festen Koordinaten auf dem Bildschirm zugeordnet. Wenn der Bildschirm berührt wird, wird ein Interrupt erzeugt, der feststellt, welche Grafik berührt wurde und eine mit dieser "Schaltfläche" verbundene Operationen auslöst.

#### Eine Heizung für die Spannungsreferenz

Die Drift der Ausgangsspannung gegenüber der Temperatur wird in ppm/°C gemessen. Bei einem Ausgangssignal von 10 V entspricht beispielsweise 1 ppm einer Spannungsdrift von 10 µV. Die Raumtemperatur in meinem Labor schwankt grob gesagt zwischen 15°C und 25°C. Wenn also der Ausgang einen Temperaturkoeffizienten von 2,5 ppm/°C aufweist, kann sich die Ausgangsspannung innerhalb dieses Temperaturbereichs um bis zu 250 µV ändern. Ich habe eine selbstgebaute Wärmekammer mit einem Temperaturregler verwendet, um diesen Koeffizienten zu bestimmen.

Dann habe ich mich gefragt, ob ich den Temperaturkoeffizienten verbessern kann, wenn ich die Spannungsreferenz aufheize. Also habe ich eine temperaturgesteuerte Heizung auf einer kleinen Leiterplatte aufgebaut und auf die Unterseite des Referenzkreises des Kalibrators geklebt. Als Sollwert wählte ich 55°C, weit oberhalb der normalen Umgebungstemperatur. Auch hier habe ich den Kalibrator in meine Wärmekammer gestellt und einen fünfstündigen Temperaturzyklus von 15°C bis 40°C durchgeführt. Ich habe eine Drift der Ausgangsspannung von nur 26 µV ermitteln können, was einen Temperaturkoeffizienten von weniger als 0,2 ppm/°C ergibt, also fünfmal besser als ohne Heizung!

Ein 3D-gedrucktes Gehäuse wurde entworfen, das die Heizung abzudecken und die Temperatur der Heizung und der Spannungsreferenz stabilisieren soll. Diese 3D-Designdateien sind ebenfalls auf der Elektor-Labs-Webseite [1] zu finden.



Bild 4. Die Heizschaltung auf der Hauptplatine unter der Spannungsreferenz.

#### Programmierung des uLCD35-Moduls

Dieser Kalibrator funktioniert nicht ohne Software. Alle benötigten Dateien stehen Sie auf der Elektor-Labs-Seite [1] zum Download bereit. Für das HMI-Modul selbst benötigen Sie das Entwicklungs-Starter-Kit SK-35DT-AR von 4D Systems, das das Touch-Display, eine 4-GB-Micro-SD-Karte und eine Adapter zur Programmierung des HMI-Moduls enthält. Dieses Kit kann bei Conrad (1307479), Digikey (1613-1050-ND) oder RS-Components (841-7790) erworben werden. Außerdem müssen Sie die kostenlose IDE von 4D System herunterladen [3] und installieren.

Um das HMI-Modul zu programmieren, vergewissern Sie sich zunächst, dass die micro-SD-Karte (maximal 4 GB) im FAT-Format formatiert ist (nicht FAT32!). Laden Sie Software.ZIP und USD\_FILES.ZIP von [1] herunter und entpacken Sie die Inhalte in Ordner auf Ihrem Computer. Kopieren Sie alle Dateien aus dem Ordner USD\_FILES auf die micro-SD-Karte und stecken Sie diese in das HMI-Modul. Schließen Sie das Programmieradapter an die zehnpolige Stiftleiste des Moduls und an einen USB-Anschluss Ihres Computers an. Öffnen Sie die Anwendung *Program Loader* (Bild 6) der 4D-System-IDE, wählen Sie den COM Port des Programmieradapters, wählen Sie Flash als Ziel, laden Sie die Datei calibrator.4XE und klicken Sie auf OK. Wenn das Programm geladen ist, zeigt das Modul alle Tasten und Encoder-Räder an.

Hinweis: Der Quellcode ist in der Datei *calibrator.4Dg* zu finden; alle Kommentare sind auf Französisch. Der Benutzer kann ihn mit der IDE bearbeiten und ändern.



Der Schaltplan der Heizungssteuerung ist in Bild 3 dargestellt. Der NTC TH1 befindet sich auf der Unterseite der Platine in Bild 4 und muss an der Spannungsreferenz U9 auf der Hauptplatine ausgerichtet und darunter festgeklebt werden. Bild 5 schließlich zeigt den Prototyp der Heizungsplatine.

#### **Bau des Kalibrators**

Die KiCad-Designdateien, die Gerber- und Bohrdateien für die Bestellung der Haupt- und der Heizungsplatine sowie die Stücklisten für beide Platinen stehen auf der Elektor-Labs-Webseite zum Download bereit [1], ebenso wie die Software zur Programmierung des HMI-Moduls. Das Projekt ist so konzipiert, dass es in das in der Stückliste angegebene Hammond-Gehäuse passt. Auch wenn das Löten von SMD-Bauteilen für unerfahrene Bastler nicht ganz einfach ist, sollte es nicht allzu schwierig sein, die Platinen mit Hilfe eines kleinen Lötkolbens zu bestücken.

#### **Batteriebetrieb**

Ich habe zur Versorgung der Kalibratorplatine zunächst ein chinesisches Akku-(Ent-) Lademodul (Ref. DDO4CVSA) des Herstellers Eletechsup



und einen 4,1-V-Lithium-Akku vorgesehen, aber nach einigen Tests wurde mir klar, dass sich die Angelegenheit zu stark erwärmte. Sie können es gerne selbst ausprobieren, natürlich auf eigene Gefahr. Beachten Sie, dass Vin und Vout an J1 mit einer Drahtbrücke verbunden werden müssen, wenn dieses Modul nicht installiert ist!

#### **Kalibrierung**

Wie jede Referenzspannungsquelle muss auch der MAX6350 künstlich gealtert werden, um die Langzeitstabilität zu verbessern. Nach 1000 Stunden, also etwa 41 Tagen Dauerbetrieb bei Raumtemperatur lag die Referenzspannung stabil bei 4,99994 V und die beiden Referenzspannungen wiesen mit 10,50050 V beziehungsweise -10,49867 V nur eine Abweichung von etwa 10  $\mu$ V auf.

Für die Kalibrierung ist dann ein Multimeter mit mindestens 6,5 Stellen erforderlich, wie sie mein Agilent 34410A aufweist. Durch Drücken und Halten der *Sign*-Taste auf dem HMI für acht Sekunden werden alle zuvor gespeicherten Kalibrierungsparameter gelöscht.

#### Kalibrierung der Ausgangsspannung

Wählen Sie zunächst durch Drücken von *Volt/mA* den Modus Spannungsausgang. Halten Sie die Taste *000* für 5 s gedrückt, um den Kalibrierungsprozess zu starten: Es ertönt ein Piepton und die Meldung *calibration mode* wird angezeigt. Mit den Tasten "+" und "-" der Encoder-Räder stellt man 0 V ( $\pm 15 \mu$ V) auf dem Multimeter ein. Warten Sie etwa 10 s, damit sich der Wert stabilisiert. Wenn Sie nun die Taste *000* drücken, wird für 2 s *cal stored* angezeigt, dann zeigen die Encoder-Räder *10.00000* an. Wählen Sie dann wieder mit den Encoder-Rädern den Wert, der erforderlich ist, damit das Multimeter 10,00000 V ( $\pm 15 \mu$ V) anzeigt.

#### Kalibrierung des Ausgangsstroms

Wählen Sie jetzt den Modus Stromausgang, indem Sie *Volt/mA* drücken. Halten Sie die Taste *000* für 5 s gedrückt, um den Kalibrierungsprozess zu starten. Es ertönt ein Piepton und die Meldung *calibration mode* wird angezeigt. Die Encoder-Räder zeigen *00.0100 mA* an. Wählen Sie mit den Tasten "+" und "-" der Encoder-Räder den richtigen Wert aus, um 10  $\mu$ A (±100 nA) auf dem Multimeter anzuzeigen. Durch Drücken der Taste *000* erscheint auf der LCD-Anzeige für 2 s *cal stored*, dann zeigen die Encoder-Räder *30.0000* an. Wählen Sie mit den Encoder-Rädern den Wert, der erforderlich ist, damit das Multimeter 30 mA (±10 nA) anzeigt.

#### WEBLINKS

- [1] Dieses Projekt auf Elektor-Labs: https://bit.ly/3hlPYHe
- [2] Das Projekt im Video: https://youtu.be/i8ZxOXWNTnU
- [3] Downloads der IDE Workshop 4:
- https://4dsystems.com.au/workshop4#downloads
- [4] H. Halloin, P. Prat und J. Brossard, "Long Term Characterization of Voltage References", arXiv.org der Cornell University, 2013: https://arxiv.org/pdf/1312.5101.pdf
- [5] L. T. Harrison, "Current Sources and Voltage References", Elsevier, 2005: https://bit.ly/3lHbavg
- [6] LTZ1000: https://xdevs.com/article/kx-ref/
- [7] AD5791 Evaluation Board: https://bit.ly/3hdrFMW



Bild 5. Die Heizungsplatine, nur SMDs...

#### **Bedienung**

Der GUI-Touchscreen zeigt berührungsempfindliche Encoder-Räder und drei Tasten. Mit den Plus- und Minus-Tasten der Encoder-Räder wird der Ausgangswert ausgewählt. Die *sign*-Taste kehrt die Polarität der Ausgangsspannung um, im Strommodus ist sie inaktiv. Die *Volt/mA*-Taste wählt den Ausgangsmodus (entweder Spannung oder Strom), und die *000*-Taste setzt den Ausgang auf 0 V beziehungsweise 0 mA zurück.

#### **Ein wertvolles Gut**

Die Gesamtkosten für den Bau dieses hochpräzisen Spannungs- und Stromkalibrators belaufen sich auf etwa 300 €, was ein wirklich guter Preis für ein solches Gerät ist. Zugegeben, Sie benötigen auch ein hochpräzises, gut kalibriertes Multimeter, um eine genaue Einstellung zu gewährleisten. Das mag ein Problem sein, aber die meisten von uns haben sicher irgendeinen Zugang zu dieser Art von Ausrüstung. Dieser Kalibrator wird Ihnen eine wertvolle Bereicherung für das Labor sein, um alle Arten von elektronischen Schaltungen, Geräten und Messinstrumenten zu testen und einzustellen.

210208-02

#### **Ein Beitrag von**

Idee, Entwurf, Text und Illustrationen: Vincent Gautier Schaltpläne: Patrick Wielders Redaktion: Luc Lemmens Übersetzung: Rolf Gerstendorf Layout: Harmen Heida

#### Haben Sie Fragen oder Kommentare?

Haben Sie Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Redakteur unter luc.lemmens@elektor.com.



#### PASSENDE PRODUKTE

> PeakTech 3442 True RMS Digital Multimeter mit Bluetooth (Anzeige 60000) (SKU 18773) www.elektor.de/18773



## Arduino Nano RP2040 Connect

#### Raspberry Pi RP2040 + WLAN + Bluetooth

Von Clemens Valens (Elektor-Labor)

Das Arduino-Board Nano RP2040 Connect stattet den Mikrocontroller RP2040 der Raspberry Pi Foundation mit WLANund Bluetooth-Funktionalität aus. Und neben diesen drahtlosen Kommunikationsarten verfügt das kleine Board über ein Mikrofon und einen sechsachsigen intelligenten Bewegungssensor mit KI-Fähigkeiten. Klingt interessant? Dann werfen wir einen Blick auf das neue Board!



**Review im Video** 

Im Januar 2021 wurde die Maker-Szene überrascht, als Raspberry Pi das Pico-Board mit dem brandneuen RP2040-Mikrocontroller vorstellte. Ein paar Monate später kündigte auch Arduino ein RP2040-basiertes Board an: Arduino Nano RP2040 Connect.

Das Board Arduino Nano RP2040 Connect macht dort weiter, wo Raspberry Pi aufgehört hat, als man sich entschied, das Pico Board nicht mit einer drahtlosen Verbindung auszustatten. Einerseits verständlich, denn das hätte den Preis in die Höhe getrieben, andererseits wurde dies von vielen Makern als verpasste Chance empfunden. Daher ist das Connect-Board mit dem WLAN- und Bluetooth-Funkmodul NINA-W102 von u-blox ausgestattet. Ein entsprechendes Modul dieses Herstellers ist auch im Arduino Nano 33 BLE zu finden.

Neben der drahtlosen Konnektivität verfügt das Connect-Board über ein Mikrofon und einen sechsachsigen intelligenten Bewegungssensor mit KI-Fähigkeiten. Eine RGB-LED ist ebenfalls verfügbar. Zweiundzwanzig GPIO-Ports, 20 Pins mit PWM-Unterstützung und sechs Analogeingänge ermöglichen es dem Benutzer, Relais, Motoren, LEDs und viele andere Aktoren zu steuern und Schalter und andere Sensoren auszulesen. Der Flash-Programmspeicher ist mit 16 MB reichlich bemessen, mehr als genug, um viele Webseiten oder andere Daten zu speichern. Das Arduino-Board Nano RP2040 Connect ist "Raspberry-Pi-kompatibel", was bedeutet, dass es nicht nur das gesamte Software-Ökosystem des RP2040 unterstützt, sondern auch MicroPython. Gleichzeitig unterstützt das Board die Arduino-Programmiersprache, die IDEs v1 und v2 und alle dazugehörigen Bibliotheken. Schließlich ist das Connect-Board vollständig kompatibel mit der Arduino-Cloud und der Smartphone-App Arduino IoT Remote.

#### **WLAN und Bluetooth**

Die drahtlose Konnektivität wird durch ein NINA-W102-Modul von u-blox bereitgestellt (**Bild 1**). Laut Datenblatt ist dieses Modul ein eigenständiges Multi-Funk-MCU-Modul, das einen leistungsstarken Mikrocontroller und ein Funkbaustein für die drahtlose Kommunikation umfasst. Es verfügt über eine offene CPU-Architektur, wie es bei u-blox heißt, was bedeutet, dass es dem User überlassen bleibt, die gesamte Software für das Modul zu schreiben. Im Auslieferungszustand tut das Modul gar nichts. Im Grunde genommen ist das NINA-W102

Bild 1. Das autonome Multi-Funk-MCU-Modul NINA-W102 von u-blox hat einen Espressif ESP32 unter der Haube. (Quelle: u-blox)







Bild 2. Überblick über die Funktionen des Arduino Nano RP2040 Connect.

ein weiteres ESP32-Modul in einer Blechumhüllung mit einer Antenne. Die Firmware für das NINA-Modul wurde vom Arduino-Team entwickelt und der Quellcode kann auf GitHub [1] eingesehen werden.

Mit diesen Erkenntnissen wundert es uns nicht mehr, dass der Arduino Nano RP2040 Connect viel leistungsfähiger ist, als wir zunächst dachten, denn er verfügt nicht nur über einen RP2040 mit zwei Cortex-M0+-Kernen, sondern auch über einen Dual-Core-ESP32 sowie über WLAN und Bluetooth. Alle WLAN- und Bluetooth-Funktionen, die normalerweise auf einem ESP32-Modul ausgeführt werden, laufen auf dem NINA-Modul, so dass der RP2040-Controller davon unbehelligt seine eigenen Aufgaben abarbeiten kann.

#### **Board-Übersicht**

Ein Blick auf das Blockdiagramm des Connect (**Bild 2**) zeigt, dass sich die beiden MCUs einen I<sup>2</sup>C-Bus teilen, an den die Inertiale Messeinheit (IMU) und das Authentifizierungselement (auch bekannt als Secure Element) angeschlossen sind, so dass beide MCUs diese nutzen können. Die beiden MCUs sind zusätzlich über einen SPI/UART-Bus

miteinander verbunden.

Die RGB-LED des Boards ist mit dem ESP32, also dem NINA-Modul, das Mikrofon mit dem RP2040 verbunden. Die Erweiterungsanschlüsse (**Bild 3**) sind ebenfalls mit dem RP2040 verbunden, mit Ausnahme von Pin A6 und Pin A7, die zum ESP32 geroutet sind. Beachten Sie, dass die analogen Pins A4 und A5 für den I<sup>2</sup>C-Bus reserviert sind; sie haben 4k7-Pull-up-Widerstände und sind mit beiden MCUs verdrahtet. Dies gilt auch für den *QSPI\_CSn*-Pin, der auf der Platine mit *REC* bezeichnet ist, im dem Pinout-Diagramm aber mit *Reset*.

#### **Die Arduino Nano-Familie**

Eigentlich scheint der Connect eine Art erweiterte oder verbesserte Version des Arduino Nano 33 IoT-Boards zu sein. Auch dieses Board vereint einen ARM Cortex-M0+ Prozessor mit einem Funkmodul NINA W102 von u-blox. Der Prozessor, ein SAMD21, ist jedoch ein Single-Core-Controller anstelle einer Dual-Core-CPU und das Board besitzt auch weder eine RGB-LED noch ein Mikrofon. Außerdem verfügt es über weniger Speicher und ist langsamer.



Bild 4. Die Arduino-Nano-Familie mit ihren vier Mitgliedern. Die Platinen messen nur 45 mm × 18 mm. (Quelle: arduino.cc)

Auf der Softwareseite ist der Connect jedoch eng mit dem Arduino Nano 33 BLE verbunden. Dieses Board besitzt ebenfalls ein NINA-Modul, aber eines, das auf einer nRF52840 MCU von Nordic Semiconductor basiert, und es gibt keine zweite MCU. Außerdem verfügt es nicht über WLAN-, sondern nur über Bluetooth-Funktionalität.

Zusammen mit dem Nano 33 BLE Sense, dem Connect und dem BLE bildet es die Mbed OS Nano Familie (Bild 4). Es ist mir nicht klar, warum der Nano 33 IoT nicht zu dieser Familie zählt (nicht genug Speicher?). Meiner Meinung nach hätte der Arduino Nano RP2040 Connect eigentlich Nano 33 Connect heißen sollen, aber das Ersetzen der 33 durch RP2040 sollte vielleicht die Verkaufszahlen steigern.

#### Software-Unterstützung

Da das Connect ein Arduino-Board ist, integriert es sich nahtlos in die Arduino-IDE. Als ich diesen Artikel schrieb, schien es jedoch keine Unterstützung für die Verwendung der beiden Kerne des RP2040 zu geben, und auch sonst gibt es nichts RP2040-spezifisches, wie etwa Unterstützung für die PIO-Blöcke oder den Interpolator. Alle Beispiele sind generische Arduino-Beispiele, die auf vielen Boards laufen. Es gibt ein nur paar Tutorials, die erklären, wie man das Mikrofon oder die IMU benutzt, aber nichts, was nur auf einem RP2040 läuft.

Das finde ich ein wenig enttäuschend, denn warum sollte man ein RP2040 auf das Board packen, wenn man keine Unterstützung dafür anbietet? Ohne diese Unterstützung wäre das Board einfach ein Nano Connect mit ähnlichen Spezifikationen und Fähigkeiten. Das RP2040 scheint hauptsächlich als I/O-Erweiterung für den ESP32 genutzt zu werden, um den Nano-Formfaktor zu erfüllen. Außerdem ermöglicht es dem Board, Mbed-OS-kompatibel zu sein, was der ESP32 nicht ist, da er kein ARM-Prozessor ist (nur ARM-basierte Boards können Mbed OS-kompatibel sein). Aber diese Ziele hätten auch durch die Verwendung irgendeiner Cortex-M7-MCU anstelle eines RP2040 erreicht werden können. Aber vielleicht hat sich diese Kritik nun, da Sie diesen Artikel lesen, schon in Wohlgefallen aufgelöst.

#### **The Fab Four**

Der Arduino Nano RP2040 Connect vereint vier große Namen der Embedded-Maker-Szene - Arduino, Raspberry Pi, ESP32 und Mbed OS - in einem einzigen Modul, was ziemlich interessant ist. Es zeigt den Willen, das Beste aus Open Source zu kombinieren, anstatt zu versuchen, alles proprietär zu halten, wie es große Halbleiterhersteller gerne tun.

#### Zusammenfassung

Zusammenfassend lässt sich sagen, dass der Connect ein kleines, aber sehr leistungsfähiges Mikrocontroller-Modul mit vielen Möglichkeiten zum Preis eines (offiziellen) Arduino Uno ist. Wenn Sie auf der Suche nach einem nicht zu teuren Board mit Wi-Fi und/oder Bluetooth sind, auf dem Mbed OS für eine IoT-Anwendung laufen kann, dann ist der Connect eine gute Option. Wenn Sie andererseits nach einer Möglichkeit suchen, mit dem RP2040 innerhalb der Arduino-IDE zu spielen, dann sind Sie wahrscheinlich mit dem viel billigeren, aber

etwas größeren Raspberry Pi Pico Board besser bedient, auf dem übrigens auch Mbed OS laufen kann. Um den Dual-Core und alle anderen RP2040-spezifischen Dinge nutzen zu können, installieren Sie in der Arduino-IDE das Boards-Paket von Earle Philhower III [2], das auch den Connect unterstützt.

210400-02

#### **Ein Beitrag von**

Text und Zeichnungen: Clemens Valens Redaktion: Jens Nickel und C. J. Abate Übersetzung: Rolf Gerstendorf Layout: Giel Dols

#### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter clemens.valens@elektor.com oder kontaktieren Sie Elektor unter redaktion@elektor.de.



#### **PASSENDE PRODUKTE**

- > Arduino Nano RP2040 Connect (SKU 19754) www.elektor.de/19754
- > Raspberry Pi Pico board (SKU 19562) www.elektor.de/19562
- > Arduino Uno (SKU 15877) www.elektor.de/15877

#### WEBLINKS .

[1] NINA-Firmware: https://github.com/arduino/nina-fw

[2] Pico/RP2040-Boards-Paket für Arduino: https://github.com/earlephilhower/arduino-pico



## Der **physische Körper** der Künstlichen Intelligenz

#### Von Tessel Renzenbrink (Niederlande)

Begriffe wie Cloud und Künstliche Intelligenz werden als Abstraktionen verstanden, die viele Menschen verwirren. In Wirklichkeit ist die Cloud "der Computer eines anderen", und KI ist in hohem Maße von Ressourcen aus der natürlichen Welt abhängig, zum Beispiel von seltenen Rohstoffen. Wenn wir sorgfältig über diese Technologien nachdenken, stellen sich viele ethische Fragen.

Irgendwo im Internet stieß ich auf diesen Hilferuf einer Mutter, die ihrem Kind in Not helfen wollte. Besagtes Kind hatte tagelang an einem Text unter Windows 365 gearbeitet, aber als es die Datei nur wenige Stunden vor dem Abgabetermin öffnete, zeigte der Bildschirm nur eine drei Tage alte Version der Arbeit, die nur aus den ersten beiden Seiten bestand und nicht aus allen sieben, die es insgesamt geschrieben hatte. Das Internet gab Tipps, wie die neueste Dateiversion zu retten wäre. Die Mutter informierte im Netz regelmäßig darüber, wie die Rettungsaktion verlief, aber es war nichts zu machen, obwohl offensichtlich drei verschiedene Microsoft-Techniker ihr beigestanden hatten. Die Datei konnte nicht wiederhergestellt werden, und die Geschichte endete mit einer letzten Meldung: Das Kind tippte verzweifelt, um den Aufsatz noch vor dem Abgabetermin um Mitternacht neu zu schreiben.

#### Die Cloud ist nur der Computer von jemand anderem

Was mich an dieser Geschichte am meisten überrascht hat, ist, dass Leute tatsächlich ihre wichtigen Dateien nur in der Cloud speichern. Ein Begriff, der sich vielleicht besser mit "der Computer eines anderen" beschreiben lässt. Da sie die Autonomie über ihre eigene Arbeit aufgegeben haben, setzen sie sich Risiken aus, wie dem Abgreifen ihrer Daten, wiederkehrenden Gebühren für den weiteren Zugriff, sich ändernden Nutzungsbedingungen oder, wie in diesem Fall, einer verpfuschten Synchronisierung. In Anbetracht der Tatsache, dass selbst die billigsten Laptops heutzutage 128 GB Speicherplatz bieten und die



Die Legende, die über KI geschaffen wird, ist die eines abstrakten und immateriellen Unterfangens Größe einer durchschnittlichen Textdatei in Kilobyte gemessen wird, gibt es aus Sicht der Nutzer keinen guten Grund, Dateien nicht lokal zu speichern. Aus Sicht des Anbieters liegen die Vorteile natürlich auf der Hand. Der Wechsel vom einmaligen Verkauf eines Produkts zum Modell X "as a Service" ist gut für die Einnahmen, erschwert die Abwanderung von Kunden und füllt die Server mit Terabytes an Daten - Daten, die analysiert werden können, um mehr über die Kundschaft zu erfahren und Erkenntnisse zu gewinnen, die zum eigenen Vorteil genutzt werden können. Die Frage, die sich stellt, ist, warum Menschen sich auf diese Art von Erpressung einlassen. Zumindest ein Teil der Antwort liegt wohl darin, dass es drei Jahrzehnte nach der breiten Einführung des Internets immer noch ein mangelndes Verständnis gibt, wie es funktioniert und welche Machtverhältnisse ihm zugrunde liegen. Und das ist kein Versehen. Der Begriff "Cloud" ist ein gutes Beispiel dafür, wie sehr das vorherrschende Narrativ über digitale Technologien deren physische und politische Realität verschleiert. Der freundlich klingende und eher vage Begriff "Cloud" weckt Assoziationen an etwas Fernes und Ungreifbare, als etwas, das Teil unseres Ökosystems ist, wir aber nicht anfassen oder (be-) greifen können. Die Formulierung "die Cloud ist nur der Computer eines anderen" gibt es schon seit langem. Doch im kollektiven Bewusstsein herrscht weiterhin die Wahrnehmung, dass massiv verteilte Datenverarbeitung etwas Flüchtiges ist. In der Zwischenzeit schreitet die Entwicklung von immer mehr und immer mächtigeren digitalen Technolo-

#### KATE CRAWFORD



Das Buch "Atlas of Al" (Quelle: Yale University Press)

gien weiter voran. Und die Erzeugung von Mythen, die sie in Geheimnisse hüllen, geht in gleichem Tempo weiter.

#### **Atlas der Kl**

In ihrem kürzlich erschienenen Buch Atlas of AI untersucht Kate Crawford den Begriff der Künstlichen Intelligenz, um zu zeigen, dass auch er diesem Akt der Verschleierung unterliegt [1]. Die Geschichte, die über KI entsteht, ist die eines abstrakten und immateriellen Unterfangens. Dieses Bild von KI als immaterieller Aktivität führt zu dem Missverständnis, dass KI nur wenige Verbindungen zu unserer physischen Welt hat, dass sie weder einen Einfluss auf die physische Welt hat noch von ihr beeinflusst wird. Crawford schreibt: "KI kann wie eine geisterhafte Kraft erscheinen - als körperlose Datenverarbeitung - aber diese Systeme sind alles andere als abstrakt. Sie sind physische Infrastrukturen, die die Erde umgestalten und gleichzeitig die Art und Weise, wie die Welt gesehen und verstanden wird, verschieben."

#### **Menschliche Arbeit hinter KI**

In Atlas of AI zeigt Crawford, dass KI von Ressourcen aus der physischen Welt wie Energie, seltenen Erden, menschlichen Interaktionen, die als Daten erfasst werden, sowie von unbezahlter und unterbezahlter menschlicher Arbeit abhängig ist. Wenn wir beim Surfen im Internet die unvermeidlichen CAPTCHAs lösen, leisten wir unbezahlte Arbeit, indem wir markierte Daten zum Trainieren von KI-Systemen erzeugen. Die Ergebnisse werden uns dann als amüsante Geschichten über künstliche Systeme verkauft, die lernen, unsere Welt ganz von allein zu verstehen. Uns wird eine nahe Zukunft versprochen, in der wir eine Unmenge an Freizeit haben werden, weil die Maschinen jede erdenkliche langweilige Aufgabe übernehmen. In Wahrheit verrichten aber Heerscharen von Menschen die sterbenslangweilige Arbeit des Beschriftens von Tausenden von Bildern über Arbeitsplattformen wie Amazon Mechanical Turk (MTurk), bei dem die Bezahlung in Cents gemessen wird.

#### Stromverbrauch

Eine weitere physikalische Komponente der KI, über die nur selten gesprochen wird, ist ihr immenser Energieverbrauch. In ihrem Buch verweist Crawford auf mehrere Forschungsarbeiten, die darauf abzielen, die für den Betrieb der Hardware erforderliche Energiemenge zu quantifizieren. Interessant ist die Studie von Strubell, Ganesh und McCullum, in der sie die Stromkosten für das Training neuronaler Netze berechnen. Sie analysierten Modelle für linguistische Datenverarbeitung wie Natural Language Processing (NLP), die für Aufgaben wie Suchmaschinen und Sprachübersetzung verwendet werden [2]. Um diese Modelle zu trainieren, muss spezialisierte Hardware wie Grafikprozessoren und Tensor Processing Units (TPUs) "durchgehend wochenoder monatelang laufen", schreiben die Autoren. In einem Beispiel berechnen sie die Kosten für die Erforschung und Entwicklung eines neuen Modells, für das etwa 60 GPUs sechs Monate lang ständig laufen mussten. Die Stromkosten betrugen 8370 € (bei einem Tarif von 0,10 €/kWh)! In ihrem Buch beschreibt Crawford viele weitere Orte und Praktiken, die zeigen, dass KI keineswegs ein rein abstraktes Unterfangen ist. "Wir brauchen eine Theorie der künstlichen Intelligenz, die die Staaten und Konzerne berücksichtigt, die sie vorantreiben und beherrschen, der Bergbau, der dem Planeten seinen Stempel aufdrückt, die massenhafte Erfassung von Daten und die zutiefst ungleichen und zunehmend ausbeuterischen Arbeitspraktiken, die sie aufrechterhält. Dies sind die tektonischen Verschiebungen der Macht in der KI." M 210415-02



Ein Kohlekraftwerk. Künstliche Intelligenz hat erhebliche Auswirkungen auf die Umwelt, zum Beispiel durch ihren steigenden Stromverbrauch. (Quelle: pxfuel.com/de/free-photo-ogrok/download)

#### WEBLINKS

 K. Crawford, Atlas of AI, Yale University Press, 2021: https://yalebooks.yale.edu/book/9780300209570/atlas-ai
 E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP", 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019: http://www.arxiv.org/abs/1906.02243



### **Projekt** 2.0 Korrekturen, Updates und Leserbriefe

Redaktion: Ralf Schmiedel, Jens Nickel

S

#### Temperaturgesteuerte Lötstation 2021 Elektor 05-06/2021, S. 30 (190409)

Bei der Ansteuerung des FET T3 (IRF9Z34NPBF) kommt es zu einer Überschreitung der zulässigen Spannungsdifferenz zwischen Gate und Source. Der T3 ist für eine maximale Spannung von 55 V zwischen Source und Drain spezifiziert, jedoch muss beim Einsatz auch die zulässige maximale Spannung zwischen Gate und Source beachtet werden, die hier 20 V beträgt. Stellt dieses beim Betrieb mit 17 V noch kein Problem dar, überschreiten die 24 V diese Grenze. Unsere aufmerksamen Leser haben kurz nach Veröffentlichung schon darauf hingewiesen, dass die Lebensspanne des FET ohne weitere Maßnahmen stark verkürzt wird. Um dies zu verhindern wird eine 15-V-Z-Diode (BZX85B15) zwischen Source und Gate eingesetzt, sowie der Widerstand R18 gegen einen 470  $\Omega/2$  W getauscht. Die Bilder zeigen die nötigen Modifikationen (Z-Diode auf der Unterseite der Platine / getauschter Widerstand).

Die Modifikationen schützen den FET T3, durch Überschreiten der Spannung zwischen Gate und Source beschädigt zu werden, erzwingen jedoch die Verwendung eines Kühlkörpers für den FET, da durch die Z-Diode und den geänderten Widerstand höhere Verluste auftreten.

Ein weiteres Problem, das nicht durch ein paar Bauteile gelöst werden kann, betrifft die Temperaturerfassung der beiden Spitzen HAKKO FX8801 und JBC T-245. Die JBC T-245 liefert nicht wie aktuell vorgesehen eine positive

Spannung für die Temperatur, sondern eine negative, welche die Station nicht verarbeiten kann. Bei der HAKKO FX8801 kommt kein Thermoelement für die Temperaturerfassung zum Einsatz, sondern ein temperaturveränderlicher Widerstand. Daher sollen diese Spitzen in Kürze durch eine Zusatzplatine abgedeckt werden, die sich in Entwicklung befindet. Wie auch bei der Station wird Wert auf THT-Komponenten gelegt, die zudem aktuell zu beschaffen sind.

Aktualisierte Gerberfiles, Schaltpläne und Stücklisten sind auf der Webseite des Originalartikels zu finden: www.elektormagazine.de/magazine/elektor-174/59555.

Mathias Claußen (Elektor-Labor)

C

ESP32 als Zeit-Server

Elektor 07-08/2019, S. 96 (180662)

Durch einen fehlerhaften Pointer konnte es beim NTP-Server in Version 1.5 zu Abstürzen kommen. Dank des Hinweises von "schnarz" (*www.elektormagazine.de/labs/mini-ntp-server-with-gps*) und des passenden Exception Dumps war das Problem zügig zu beheben. Version 1.6 erlaubt damit endlich die Nutzung der verbesserten Synchronisierung. Mit Version 1.6 wurde das Projekt auch nach PlatformIO konvertiert, so dass Bibliotheken nun automatisch passend heruntergeladen werden können. Version 1.6 ist auf GitHub zu finden (*https://github.com/ElektorLabs/180662-mini-NTP-ESP32*).

Es freut uns, dass dieses Projekt durch unsere Leser nicht nur nachgebaut, sondern auch verbessert wird! Mathias Claußen (Elektor-Labor)





#### Solaranlage für Mähroboter

Elektor 07-08/2021, S. 105 (200553)

In die Schaltung des Spannungswandlers in Bild 9 hat sich ein Fehler eingeschlichen. Die Anode von

D1 muss natürlich an die Verbindung zwischen L1 und Drain von T1. C2 puffert den Eingang (siehe Bild). Auf die Beschreibung im Text hat das keinen Einfluss, denn da ging es lediglich um die rot gezeichneten Änderungen; die feste Einstellung auf 27 V am Ausgang und die Strombegrenzung.





LoRa mit dem Rasperry Pi Pico Elektor 07-08/2021, S. 06 (210047)

Zum Status der Platine kamen einige Rückfragen und Anregungen. Die Platine ist noch nicht bestellt

oder aufgebaut und daher auch leider nicht getestet worden. Wünsche und Anregungen können aber gerne mit uns auf einer neuen Elektor-Labs-Seite zum Projekt geteilt werden:

www.elektormagazine.com/labs/ raspberry-pi-pico-with-lora-and-recharable-lithium-battery Mathias Claußen (Elektor-Labor)

#### Möchten Sie selbst etwas beitragen?

Gerne können Sie sich an die Elektor-Redaktion wenden unter der E-Mail-Adresse: redaktion@elektor.de



#### Spielen mit SC/MP

#### Elektor 06/1977, S. 28 (9846)

Für mich, und ganz sicher auch für viele andere Leser, war der Artikel "Spielen mit SC/MP" der prägendste Artikel, der jemals in Elektor veröffentlicht wurde.

Mikroprozessoren waren damals ganz neu, jeder sprach darüber. Aber ganz ehrlich, die Wenigsten von uns wussten wirklich, wie diese neuen Bauteile funktionieren und was damit machbar ist. Dieser Artikel änderte dies schlagartig und weckte die Neugier und für viele die Leidenschaft.

Ich hatte damals gerade mein Studium der Nachrichtentechnik begonnen, wollte aber unbedingt wissen, was es mit diesen Mikroprozessoren auf sich hat. Zusammen mit zwei Kommilitonen begannen wir, den SC/MP aufzubauen. Die Begeisterung und Leidenschaft war spätestens geweckt, als wir das erste Mal "ELBUG.." auf dem Display lasen. Wir änderten unsere Studienrichtung auf "Informationstechnik" - diese Leidenschaft hat unser gesamtes Berufsleben geprägt. Auf dem Speicher habe ich das System von damals gefunden und in Betrieb genommen. Es lebt wieder!

**Roland Stiglmayr** 





# Grafische Benutzeroberflächen **mit Python und guizero**

Installieren Sie die Python-Bibliothek guizero und erstellen Sie Ihre eigenen grafischen Benutzeroberflächen.

Mit einer grafischen Benutzeroberfläche (GUI) lassen sich Python-Programme leichter bedienen und interessant gestalten.



Laura leitet das A-Level-Team bei der Raspberry Pi Foundation und erstellt Ressourcen für Studenten, die etwas über Informatik lernen

#### @ CodeBoom

wollen.



MAKER

Martin arbeitet im Lernteam der Raspberry Pi Foundation, wo er Online-Kurse, Projekte und Lehrmaterial erstellt.

O'Hanlon

@ martinohanlon

 Eine alternative Möglichkeit, guizero zu installieren, besteht darin, die Zip-Datei von GitHub herunterzuladen. Sie können verschiedene Komponenten, sogenannte "Widgets", zu Ihrer Oberfläche hinzufügen, um Informationen in das Programm einzugeben und auf dem Bildschirm darzustellen. In dieser Serie wenden wir uns der guizero-Bibliothek zu, die gerade auch für Anfänger gut geeignet ist. Das Standard-GUI-Paket von Python heißt *tkinter*, ist auf den meisten Plattformen bereits mit Python installiert und kann mit der guizero-Bibliothek auf einfache Weise genutzt werden.

#### 01 Installation von guizero

Um die gezeigten Programme zu erstellen, müssen Sie die Python-Bibliothek guizero (**lawsie.** github.io/guizero) installieren. Sie ist als Python-Paket verfügbar, also als wiederverwendbarer Code, den Sie herunterladen, installieren und dann in Ihren Programmen verwenden können. Wie Sie guizero installieren, hängt von Ihrem Betriebssystem und Ihren Administratorrechten ab. Wenn Sie Zugriff auf die Kommandozeile oder ein Terminal haben, können Sie den folgenden Befehl verwenden:

pip3 install guizero

Unter lawsie.github.io/guizero finden Sie umfassende



Installationsanweisungen für guizero und Optionen für die Installation, falls Sie keine Administrator-Rechte für Ihren Computer haben. Sie finden dort auch Installationen für Windows zum Download.

#### 02 Hello World

Wenn Sie guizero installiert haben, überprüfen Sie, ob es funktioniert: Schreiben Sie eine kleine "Hello World"-App, die traditionell als erstes Programm beim Ausprobieren einer neuen Sprache geschrieben wird. Öffnen Sie den Editor, in welchem Sie Ihren Python-Code schreiben werden. Am Anfang eines jeden guizero-Programms wählen Sie die benötigten Widgets aus der guizero-Bibliothek aus und importieren sie diese. Sie brauchen jedes Widget nur einmal zu importieren und können es dann beliebig oft in Ihrem Programm verwenden. Fügen Sie oben auf der Seite den folgenden Code ein, um das App-Widget zu importieren:

from guizero import App

Alle guizero-Projekte beginnen mit einem Hauptfenster (*App* gnannt), das ein Container-Widget ist. Am Ende jedes guizero-Programms müssen Sie dem Programm sagen, dass es die App anzeigen soll, die Sie gerade erstellt haben. Fügen Sie diese beiden Codezeilen unterhalb der Zeile ein, in der Sie das App-Widget importiert haben:

app = App(title="Hello world")
app.display()

Speichern Sie nun Ihren Code und führen Sie ihn aus. Sie sollten ein GUI-Fenster mit dem Titel "Hello World" sehen (**Bild 1**). Herzlichen Glückwunsch, Sie haben soeben Ihre erste guizero-App erstellt!





Bild 1 Ihre erste guizero-App

#### Widgets 03

Widgets sind die Dinge, die auf der Benutzeroberfläche erscheinen, wie z. B. Textfelder, Schaltflächen, Schieberealer und sogar einfache Textstücke. Alle Widgets werden zwischen der Codezeile zum Erstellen der App und der Zeile app.display() eingefügt. Hier ist die App, die Sie gerade erstellt haben, aber in diesem Beispiel haben wir ein Text -Widget hinzugefügt:

```
from guizero import App, Text
app = App(title="Hello world")
message = Text(app, text="Welcome to the app")
app.display()
```

Haben Sie bemerkt, dass es zwei Änderungen gibt (Bild 2)? Es gibt jetzt eine zusätzliche Codezeile, um das Text-Widget hinzuzufügen, und wir haben auch Text zur Liste der zu importierenden Widgets in der ersten Zeile hinzugefügt.

Schauen wir uns den Code des Text-Widgets etwas genauer an:

message = Text(app, text="Welcome to the app")

Wie jede Variable in Python braucht auch ein Widget einen Namen. Dieser heißt hier "message". Dann geben wir an, dass dies ein "Text"-Widget sein soll.

Innerhalb der Klammern stehen einige Parameter, die das Aussehen des Text-Widget bestimmen. Der erste Parameter, "app", teilt dem Text mit, wo sich das Widget befindet. Alle Widgets müssen sich innerhalb eines Container-Widgets befinden. Die meiste Zeit werden Ihre Widgets direkt in einer App enthalten sein, aber es gibt, wie Sie noch sehen werden, auch einige andere Arten von Container-Widgets, in die Sie Dinge hinein stecken können. Schließlich weisen wir das Widget an, den Text "Welcome to the app" darzustellen.

#### Ein Fahndungsplakat

#### **Schriftart- und Farbenvielfalt** 01

Nachdem Sie nun eine Benutzeroberfläche erstellen können, wollen wir sie etwas interessanter gestalten. Sie können Text in verschiedenen Schriftarten, Größen und Farben hinzufügen, die Hintergrundfarbe ändern und auch Bilder hinzufügen. Um das alles zu üben, wollen wir ein "Wanted"-Plakat nach traditioneller Wild-West-Manier erstellen. Als Erstes müssen Sie eine App erstellen. Fügen Sie in Ihrem Editor diesen Code ein, um ein einfaches App-Fenster zu erstellen:

from guizero import App

app = App("Wanted!")

app.display()

Wenn Sie den Code speichern und ausführen, sollten Sie eine Grafik sehen, die wie ein einfaches graues Quadrat mit dem obenstehenden Titel "Wanted!" aussieht (Bild 3).

#### Hintergrundfarben 02

Lassen Sie uns den Hintergrund der App ein wenig ändern. Fahndungsplakate sehen meist so aus, als seien sie aus vergilbtem Pergament. Ändern wir die Hintergrundfarbe daher in Blassgelb.

Suchen Sie die Code-Zeile, in der Sie die App erstellen. Unmittelbar nach dieser Codezeile fügen Sie eine weitere Codezeile ein, um die Eigenschaft bg (background) des Fensters zu ändern. Mit dieser Eigenschaft können wir die Farbe des Hintergrunds ändern:

#### Alles. was Sie brauchen

- > Einen Computer (z. B. Raspberry Pi, AppleMac, Windows- oderLinux-PC)
- > Internetverbindung
- > Python 3 (python.org)
- > Eine IDE (Code-Editor), z. B.: IDLE (wird mit Python 3 installiert). Thonny (thonny.org), Mu (codewith.mu),PyCharm (jetbrains.com/pycharms)
- > Die guizero Python library (lawsie.github.io/guizero)





from guizero import App
app = App("Wanted!")
app.bg = "yellow"

app.display()

Dies wird als Editieren einer Eigenschaft bezeichnet. Im Code müssen Sie das Widget angeben, um das es geht (app), die Eigenschaft, die Sie ändern möchten (bg) und den Wert, auf den Sie sie ändern möchten. Vielleicht ist diese Farbe (**Bild 4**) ein bisschen zu knallig. Lassen Sie uns daher den Hex-Code einer helleren gelben Farbe nachschlagen. Es gibt viele Websites, auf denen Sie nach Farben suchen können, zum Beispiel auf **htmlcolorcodes.com (Bild 5**).

Wenn Sie die gewünschte Farbe ausgewählt haben, wird ihr Code auf der Website entweder als Hexadezimalwert (in diesem Fall #FBFBD0) oder als RGB-Wert (251, 251, 208) angezeigt. Sie können beide Formate zum Einstellen von Farben in *guizero* verwenden:

app.bg = "#FBFBD0" app.bg = (251, 251, 208)

#### 03 Hinzufügen von Text

Ihre App sollte ungefähr so aussehen wie in Bild 6. Fügen wir nun etwas Text zur GUI hinzu: Den typischen Fahndungsplakat-Schriftzug: "Wanted"!



Bild 4	Wanted!	~ ^ X
Bild 4 Andern der		

Suchen Sie zunächst nach der Codezeile, in der Sie die App bereits importiert haben.

from guizero import App

Um einen Text erstellen zu können, fügen Sie "Text" Ende der Liste an. Jetzt sieht die Zeile wie folgt aus:

from guizero import App, Text

Jedes Mal, wenn Sie einen neuen Widget-Typ verwenden möchten, fügen Sie dessen Namen an das Ende der Liste an. Es gibt keinen Grund, ständig neue Codezeilen hinzuzufügen: Bleiben Sie einfach bei einer Liste, damit Ihr Programm nicht zu unübersichtlich wird. Da Sie nun Text verwenden können, fügen wir ein Stück Text hinzu. Denken Sie daran, dass alle Widgets auf der

bg ist die Abkürzung für 'background' und erlaubt uns, die Farbe des Hintergrunds zu ändern 20

#### guizero-Dokumentation

Auch wenn Sie Programmieranfänger sind, können Sie die volle Leistungsfähigkeit von guizero und allen anderen Bibliotheken, denen Sie begegnen, nutzen.

Die guizero-Dokumentation finden Sie unter **lawsie**. github.io/guizero. Wenn Sie zum Beispiel mehr über das Ändern von Widget-Properties erfahren möchten, klicken Sie dort auf das Widget, das Sie ändern möchten, und scrollen Sie nach unten, bis Sie den Abschnitt mit den Eigenschaften erreichen. Wenn Sie z. B. "Text" unter der Überschrift "Widgets" auswählen, werden alle editierbaren Eigenschaften eines Textes angezeigt. Die Dokumentation enthält oft hilfreichen Code, der Ihnen zeigt, wie Sie eine bestimmte Eigenschaft oder Methode verwenden können.



Bild 7

Bild 7 Der Text ist zu klein

Bild 6 Blassgelber Hintergrund

GUI zwischen der Codezeile, in der Sie die App erstellen, und der Codezeile, in der Sie sie anzeigen, eingefügt werden müssen. Ihr Code sollte jetzt so aussehen:



Schauen wir uns die gerade hinzugefügte Code-Zeile einmal genauer an.

#### wanted\_text = Text(app, "WANTED")

Hier ist wanted\_text der Name des Textes, damit wir später im Code darüber sprechen können - stellen Sie sich das wie den Namen einer Person vor. (Sie können Ihr Textstück auch Dave nennen -dem Computer ist das egal!) Innerhalb der Klammern haben wir zwei Ausdrücke. Der zweite, WANTED, ist einfach, da es der Text ist, den wir auf dem Bildschirm anzeigen möchten. Das erste ist der Container, der dieses Stück Text steuert und Master genannt wird. In diesem Fall sagen wir, dass dieser Text von der App gesteuert werden soll. Wenn Sie zum ersten Mal mit dem Erstellen von GUIs beginnen, werden die meisten Ihrer Widgets die App als ihren Master haben, aber es gibt noch andere Container, die Widgets speichern können.

#### Änderung von Textgröße 04 und -Farbe

Die Buchstaben sind ziemlich klein (Bild 7). Ändern wir die Eigenschaft text\_size auf genau dieselbe Weise, wie wir es getan haben, als wir die Hintergrundfarbe der App geändert haben. Erinnern Sie sich, dass Sie drei Dinge angeben mussten:

1. Den Namen des Widgets

- 2. Welche Eigenschaft geändert werden soll
- 3. Der neue, geänderte Wert

In diesem Fall geben Sie also das Widget (wanted\_ size), die zu ändernde Eigenschaft (text\_size) und den neuen Wert (50) an. Fügen Sie eine neue Codezeile direkt unter der Zeile ein, in der Sie den Text erstellt haben, um die Eigenschaft zu ändern.

WANTED

wanted\_text = Text(app, "WANTED") wanted\_text.text\_size = 50

Sie haben jetzt einen Text mit größeren Buchstaben auf Ihrem Plakat (Bild 8), Versuchen Sie nun, die Schriftart dieses Textes zu ändern. Welche Schriftarten zur Verfügung stehen, hängt davon ab, welches Betriebssystem Sie verwenden, daher hier einige Vorschläge:

- Times New Roman
- Verdana
- Courier
- Impact

Bild 8

Kein Fahndungsplakat wäre vollständig ohne ein Bild, also fügen wir eines hinzu. Bild 9 zeigt die Katze des Autors, weil sie ständig an Möbeln und Tapeten kratzt. Speichern Sie eine Kopie des Bildes, das Sie verwenden möchten, im gleichen Ordner wie Ihr GUI-Programm. Sie können Bilder in anderen Ordnern verwenden, aber dann müssen Sie den Pfad zum Bild angeben, daher ist es viel einfacher, sie am Anfang einfach im gleichen Ordner zu speichern.

Denken Sie daran, dass Widges immer am Anfang des Programms importiert werden müssen. Dann folgt ein Widget mit einem sinnvollen Namen nach der Codezeile, in der Sie die App erstellen, aber vor der abschließenden app.display()-Zeile.





Benutzeroberflächen mit Python erstellen

Weitere Anleitungen zum Erstellen eigener GUIs mit guizero finden Sie in unserem neuen Buch Create Graphical User Interfaces with Python. Auf 156 Seiten finden Sie alle wichtigen Informationen und eine Reihe von interessanten Projekten. magpi.cc/ pythongui

Bild 8 Größerer Text



#### Bild 9

Bild 9 Das voll-

ständige Poster.



Fügen Sie ,Picture' zur Liste der Widgets hinzu, die beim Programmstart importiert werden sollen.

from guizero import App, Text, Picture

#### Bildbearbeitung

Da guizero eine Bibliothek für Einsteiger ist, enthält sie keine ausgefeilten Bildbearbeitungsfunktionen. Dazu ist eine zusätzliche Bibliothek namens "pillow" erforderlich. Sie können nicht-animierte GIF-Bilder auf jeder Plattform und PNG-Bilder auf allen Plattformen außer Mac verwenden. Wenn Sie also nicht sicher sind, ob Sie die zusätzlichen Bildbearbeitungsfunktionen installiert haben, bleiben Sie bei diesen Bildtypen.

Erstellen Sie nun ein Picture-Widget mit zwei Parametern: der App und dem Dateinamen des Bildes. Dies ist der Code, den wir verwendet haben, weil unser Bild **tabitha.png** heißt.

cat = Picture(app, image="tabitha.png")

Führen Sie Ihren Code, der dem auf der letzten Seite ganz unten (**02-wanted.py**) ähneln sollte, erneut aus. Das Bild unter Ihrem Text wird jetzt angezeigt (**Bild 9**). Jetzt können Sie Ihre neu erworbenen Kenntnisse nutzen, um Ihre grafischen Oberflächen nach Ihren Wünschen zu gestalten. ►

210419-0X

Anmerkung: Dieser Artikel erschien zuerst im MagPi Magazin. Elektor gibt MagPi in niederländischer, französischer und deutscher Sprache heraus.



004. app.bg = "#FBFBD0" 005. 006. wanted\_text = Text(app, "WANTED") 007. wanted\_text.text\_size = 50 008. wanted\_text.font = "Times New Roman" 009. cat = Picture(app, image="tabitha.png") 010. 011. app.display()

#### **JETZT ABONNIEREN UND SIE ERHALTEN**



# CO2-Messgerät-Set fürs Klassenzimmer

Ein ESP8266-basiertes Gerät, entwickelt von der FH Aachen

Von Thomas Dey, Ingo Elsen, Alexander Ferrein, Tobias Frauenrath, Michael Reke und Stefan Schiffer (Fachhochschule Aachen)

Da viele Kinder nicht geimpft werden können, sind nach den Sommerferien die Corona-Infektionszahlen gerade in dieser Altersgruppe sprunghaft angestiegen. Um so wichtiger ist eine regelmäßige und ausreichende Belüftung der Klassenzimmer. Die CO<sub>2</sub>-Konzentration in der Raumluft ist ein guter Indikator für die Luftqualität. Lehrer an der Fachhochschule in Elektors Heimstatt Aachen haben deshalb ein CO<sub>2</sub>-Messprojekt gestartet. Die Schüler können das Gerät mit vorgefertigten Elektronikmodulen selbst bauen.





Bild 1. Das fertig aufgebaute Gerät mit Ampel und Display in einem Kartongehäuse.

Das Umweltbundesamt (BMU) informiert auf seiner Website, dass regelmäßiges Lüften hilft, die Aerosolbelastung zu reduzieren [1]. Darüber hinaus wird empfohlen, CO<sub>2</sub>-Ampeln zu verwenden, um einen Indikator dafür zu haben, wann die Luft in einem Raum "verbraucht" ist und gelüftet werden sollte. Viele Schulen sind daher dabei, Messgeräte anzuschaffen. Das Problem ist jedoch, dass viele handelsübliche Geräte aufgrund der gestiegenen Nachfrage nicht mehr oder erst später erhältlich sind.

In den Maker-Communities im Internet kursieren Bauanleitungen, die zeigen, wie man mit elektronischen Bauteilen, die auf dem Markt erhältlich sind, eine CO<sub>2</sub>-Ampel bauen kann. Unabhängig davon hat sich eine Projektgruppe an der Fachhochschule Aachen zusammengefunden, um einen solchen Bausatz zu entwickeln. Dabei hat sie verschiedene Sensoren in unterschiedlichen Preisklassen untersucht und bewertet. Mit diesen Ergebnissen hat ein Forscherteam des Fachbereichs Elektrotechnik und Informationstechnik ein hochwertiges CO2-Messgerät entwickelt, das kostengünstig ist und auf vorrätige Materialien zurückgreift [2]. Dieses Messgerät soll in Klassenräumen eingesetzt werden, ist aber natürlich auch für andere Innenräume geeignet. Das CO<sub>2</sub>-Messgerät wird als praktisches Kit geliefert. Die Anleitung steht den Lehrern als Online-Tutorial zur Verfügung, so dass das Messgerät gemeinsam mit den Schülern aufgebaut und besprochen werden kann. Das Ganze ist dann für die Schulen zum Selbstkostenpreis von etwa





Bild 2. Fritzing-Diagramm der Module, mit dem ESP8266 als Gehirn des Projekts.



Bild 3. Unsere CO<sub>2</sub>-Messgeräte können ihre Daten über das MQTT-Protokoll [8] an das Netzwerk senden.

65 € erhältlich.

Entwickelt und betreut wird das Projekt im neuen "Makerspace" des Fachbereichs Elektrotechnik und Informationstechnik [3]. Dieser Raum soll es Wissenschaftlern und Studenten ermöglichen, ihre eigenen Ideen praktisch umzusetzen. Hier gibt es Arbeitsplätze, Materialien und Geräte, zu denen zum Beispiel ein 3-D-Drucker gehört.

#### **Der Bausatz**

Die Hauptbestandteile des  $\rm CO_2$ -Meter-Kits sind:

- > CO<sub>2</sub>-Sensor Sensirion SCD30
- Mikrocontroller-Platine NodeMCU ESP8266-12F
- > OLED, 1,3",128 x 64 Pixel mit SH1106 Controller
- > LED-Ampelmodul
- > Echtzeituhr DS3231

Das Kit enthält auch Wago-Klemmen und Jumper-Drähte sowie ein USB-Steckernetzteil und ein Micro-USB-Kabel. Ein einfaches Gehäuse kann aus einem "vorkonfigurierten" Karton hergestellt werden, der ebenfalls im Bausatz enthalten ist.

**Bild 1** zeigt das fertige Gerät mit der LED-Ampel. Wir haben uns dafür entschieden, einen Bausatz anzubieten, der kein Löten erfordert, um den Einstiegspreis so niedrig wie möglich zu halten. Auf diese Weise wird außer einem Cuttermesser kein weiteres Werkzeug benötigt, um das Gerät aus dem Bausatz fertigzustellen. Anleitungen finden Sie unter [2] und [4].

#### **Das Gehirn des Projekts**

**Bild 2** zeigt die Verdrahtung der genannten Module. Der beliebte ESP8266 ist das Gehirn des Projekts; seine WLAN-Fähigkeiten wurden in früheren Elektor-Ausgaben bereits mehrfach besprochen. Die drahtlose Schnittstelle kann genutzt werden, um die Daten der  $CO_2$ -Konzentration über das MQTT-Protokoll an offene Cloud-Plattformen im Internet zu senden, die weltweit zugänglich sind. Dies ermöglicht auch die Verwaltung eines ganzen Satzes von  $CO_2$ -Messgeräten zum Beispiel in Schulen **(Bild 3)**.

Auf der anderen Seite betreibt der ESP8266-Mikrocontroller einen Webserver und bietet Webseiten an, die in einem beliebigen Browser angezeigt werden. Über diese Webseiten kann der Benutzer das Gerät konfigurieren. Viele Elektor-Leser werden





Bild 5. Die Klemmen werden verwendet, um die SDA- und SCL-Signale von I<sup>2</sup>C und die 5 V zur Stromversorgung der Module zu verteilen.

Bild 4. Eine der Konfigurationswebseiten. Benutzer können die Schwellwerte für die Ampel-LEDs eingeben. Auch eine englische Version der Webseiten ist verfügbar.

wissen, dass der ESP8266 sowohl ein eigenes WLAN aufbauen als auch sich in bestehende Netzwerke einloggen kann. Dieses Gerät kann in beiden Modi betrieben werden, die vom Benutzer über den Webbrowser konfiguriert werden. Zusätzlich lassen sich (neben vielen anderen Parametern) die CO2-Schwellwerte für die drei LEDs der Ampel einstellen (Bild 4). Um das Design modular und einfach zu halten, verwenden wir ein NodeMCU-Board als Breakout-Board für den ESP8266-Controller. Sein USB-Anschluss macht die (Re-) Programmierung einfach. Das NodeMCU-Board kann über USB mit Strom versorgt werden. Das Board erzeugt nicht nur 3,3 V für den ESP8266, es stellt auch beide Spannungen, 5 V und 3,3 V, zur Versorgung der anderen Module auf Klemmen zur Verfügung.

#### **Peripherie-Module**

Der ESP8266 kommuniziert über I<sup>2</sup>C mit dem CO2-Sensor (mehr über den Sensor im nächsten Abschnitt). An denselben I<sup>2</sup>C-Bus haben wir auch eine Echtzeituhr angeschlossen, die von einer CR2032-Knopfbatterie am Laufen gehalten wird, um Zeitstempel für unsere Sensordaten zu generieren. Wenn schon I<sup>2</sup>C, dann verwenden wir auch ein Display mit dieser Schnittstelle, um die Verdrahtung einfach zu halten. Wie der Fritzing-Schaltplan in Bild 2 zeigt, benutzen wir Klemmverbindungen, um die SDA- und SCL-Signale des I<sup>2</sup>C-Busses sowie die 5 V für die Stromversorgung der Module vom NodeMCU-Board zu verteilen (siehe Bild 5). Eine Ausnahme ist die RTC, da sie mit 3,3 V versorgt werden muss. Die RTC-Module, die wir verwenden, werden mit nicht aufladbaren Batterien geliefert. Auf dem Board aber sorgen eine einfache 1N4148-Diode in Reihe mit einem 200-Ω-Widerstand für einen kontinuierlichen Ladestrom, was nur bei wiederaufladbaren Batterien sinnvoll ist. Nicht wiederaufladbare Batterien können explodieren, wenn man versucht, sie zu laden! Da die Lebensdauer der Batterie die erwartete Lebensdauer der Sensoren übersteigt, haben wir beschlossen, die Diode auf der RTC zu entfernen, um den gefährlichen Ladestromkreis zu deaktivieren.

Wie gesagt, werden drei LEDs in Form einer Ampel verwendet, um einen schnellen Hinweis auf den CO2-Zustand zu erhalten. Diese LEDs sind mit drei GPIOs des ESP8266 verbunden, die als Ausgang konfiguriert sind. Wir haben uns für ein komplett fertiges Ampelmodul entschieden, in dem die notwendigen LED-Vorwiderstände bereits integriert sind.

#### **Der CO<sub>2</sub>-Sensor**

Die wichtigste Komponente ist natürlich der Sensor selbst: Um Luftqualität zu messen, gibt es eine Reihe von Sensoren auf dem Markt, die im Grunde alle einem von zwei Messprinzipien folgen: (a) VOC und (b) NDIR.

VOC steht für Volatile Organic Compounds (flüchtige organische Verbindungen) und beruht darauf, dass Gase wie Kohlenmonoxid, Kohlendioxid oder Benzol die Leitfähigkeit des Sensors verändern. NDIR steht für Non-Dispersive InfraRed. Grob gesagt, wird die Kohlendioxidkonzentration in einem Messrohr durch die Intensität des Infrarotlichts abgeschätzt, die von dem Gasgemisch im Rohr reflektiert wird.

Wir haben eine Reihe von Sensoren beider Klassen getestet, indem wir eine Plastikbox mit einer bestimmten Menge CO<sub>2</sub> füllten (siehe Bild 6). Schon gleich zu Beginn unserer Experimente wurde deutlich, dass VOC-Sensoren für den Einsatz im Klassenzimmer nicht gut geeignet sind. Sie geben zwar einen groben Hinweis auf die Luftqualität, aber wir wollten möglichst genaue Konzentrationen ermitteln, um eine Empfehlung für die Belüftung des Klassenzimmers geben zu können. Außerdem wollen wir die Ergebnisse aus verschiedenen Räumen vergleichen, um Schlussfolgerungen ziehen zu können. Aus diesem Grund haben wir uns grundsätzlich für einen nicht-dispersiven Infrarotsensor in unserem CO<sub>2</sub>-Messgerät entschieden.

Dann haben wir eine Anzahl verschiedener





Bild 6. Test verschiedener Sensoren mit bestimmten CO<sub>2</sub>-Konzentrationen.



Bild 7. Vergleich der CO2-Sensoren MH-Z19 und SCD-30 [8].

NDIR-Sensoren verglichen, die auf dem Markt erhältlich sind. Insbesondere haben wir Versuche mit dem MH-Z19B von Winsen, dem S8 von Sensair und dem SCD-30 von Sensirion durchgeführt, um einen Eindruck zu gewinnen, wie genau, stabil und reproduzierbar die verschiedenen NDIR-Sensoren messen und wie groß die Abweichungen zwischen verschiedenen Chargen der gleichen Sensoren sind.

Bild 7 zeigt einen Vergleich zwischen einem

MH-Z19 und einem SCD-30 in einem realen Szenario. Es ist zu sehen, dass beide Sensoren in diesem Szenario qualitativ die gleiche  $CO_2$ -Konzentration messen. Auch die Abweichungen zwischen den Sensoren eines Typs lagen in einem akzeptablen Bereich. Allerdings haben wir nur Stichproben genommen und sind weit davon entfernt, Aussagen aufgrund systematischer und groß angelegter Testreihen zu machen.

Letztendlich haben wir uns, auch unter

#### Partnerschaft mit Elektor

Obwohl das Projekt bereits abgeschlossen ist und gut läuft, gibt es noch Dinge, die verbessert werden könnten. Ein Student der Fachhochschule Aachen und Mathias Claussen vom Elektor-Labor arbeiten derzeit an der Software, um das Gerät noch modularer und portabler zu machen. Damit wird es möglich, Komponenten des Projekts auszutauschen - zum Beispiel den ESP8266 durch den leistungsfähigeren ESP32 zu ersetzen. Anstelle des derzeit verwendeten CO2-Sensors könnten wir auch andere Typen verwenden.

Was die Pandemie betrifft, so ist es sicher kein Zufall, dass das Elektor-Labor auch an einer anderen CO<sub>2</sub>-Messplattform arbeitet, in diesem Fall auf der Basis eines ESP32 und mit Platinen zum bequemen Aufbau des Geräts. Wenn die Software vollständig portabel ist, wird es möglich sein, sie sowohl für CO<sub>2</sub>-Meter-Projekte als auch für andere Umweltsensoren zu verwenden, die meist sehr ähnliche Anforderungen und Dienste abdecken müssen. Achten Sie auf die Artikel in den kommenden Ausgaben unserer Zeitschrift!

Berücksichtigung der Lebensdauer des Sensors, für den SCD-30 von Sensirion in unseren Geräten entschieden. Ein Datenblatt findet sich auf der Sensirion-Website [5].

#### Software

Das Zustandsdiagramm in **Bild 8** zeigt den Zyklus des Softwaresystems des Geräts: Er sieht eine Startphase vor, in der der Benutzer das Gerät konfigurieren kann, eine Netzwerkstartphase, in der gegebenenfalls Netzwerkeinstellungen vorgenommen werden, und eine Messphase, in der die CO<sub>2</sub>-Konzentrationen gemessen und angezeigt werden.

Nach einer Geräteinitialisierung öffnet das Gerät in der Startphase einen Zugangspunkt, über den der Benutzer eine Verbindung zum Gerät herstellen und konfigurieren kann. Der Benutzer kann dann über einen Webbrowser auf das Gerät zugreifen und eine Reihe von weiteren Konfigurationseinstellungen ändern, zum Beispiel den Gerätemodus, die WLANund die MQTT-Einstellungen, die interne Uhr einrichten oder die Sensorkalibrierung starten. Die Einstellungen werden dauerhaft im Flashspeicher des Geräts gespeichert.

Wenn sich in dieser Phase kein Benutzer mit dem Gerät verbindet, geht das Gerät nach einer Wartezeit von etwa 30 s in die Messphase über. Je nach Gerätemodus stellt das Gerät eine Verbindung zu einem externen WLAN-Access-Point her oder verbindet sich mit dem angegebenen MQTT-Server.



Bild 8. Der in der Software implementierte Zustandsautomat.

Das Messintervall des Geräts beträgt 5 s: Jeder neue Datenpunkt wird auf dem OLED-Display des Geräts angezeigt, das entsprechende Licht der Ampel-LED wird gesetzt und die Datenwebseite des Geräts wird aktualisiert. Durch das Anbieten einer MQTT-Schnittstelle ist es besonders einfach, die Daten weiterzuverarbeiten und zum Beispiel in einer Datenbank zu speichern, die Daten mit Tools wie Node-RED darzustellen oder sogar Cloud-Dienste wie Grafana dafür zu nutzen.

Die Software ist als Rohdatei unter [4] zu finden, der Sourcecode ist stets aktuell auf GitHub [6] verfügbar.

#### **Status und Ausblick**

Schulen, aber auch Privatpersonen, die sich für einen Bausatz interessieren, finden unter [7] einen Ansprechpartner. Mehrere hundert CO<sub>2</sub>-Messgeräte sind bereits an deutschen Schulen und in Kindergärten im Einsatz. Einige dieser Messgeräte nehmen an einer Studie zur Messung der Luftqualität teil. Dort möchten wir verschiedene Unterrichtsszenarien, die hinsichtlich der Luftqualität besonders problematisch sind, erfassen und wissenschaftlich auswerten.

Darüber hinaus arbeiten die Autoren und weitere Mitglieder des Teams derzeit an einer batteriebetriebenen Version der Hardware, die

#### CO<sub>2</sub>-Konzentration

der Atmosphäre beträgt etwa 400 ppm. 4% Kohlendioxid (etwa 40.000 ppm) ab, das sich in der Raumluft verteilt. Wenn sich Menschen in einem Raum ohne Lüftung aufhalten, erhöht sich zwangsläufig die Angaben des Umweltbundesamtes gelten CO<sub>2</sub>-Werte bis 1000 ppm noch als gut, Werte über 2000 ppm als schlecht.

auf einer kleinen Platine aufgebaut werden soll. Ziel ist es, die Sensorik dann in eine intelligente Gebäudeinfrastruktur zu integrieren. 🖊

210388-03

#### Haben Sie Fragen oder **Kommentare?**

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an die Autoren unter co2metergbr@gmail.com oder an die Redaktion unter redaktion@elektor.de.

#### **Ein Beitrag von**

Text und Bilder: Thomas Dey, Ingo Elsen, Alexander Ferrein, Tobias Frauenrath, Michael Reke und Stefan Schiffer Redaktion: Jens Nickel Übersetzung: Rolf Gerstendorf Layout: Harmen Heida

#### **WEBLINKS**

- [1] Informationen über Aerosole in Räumen: https://bit.ly/2W1KFFX
- [2] Weitere Informationen über das Projekt: https://maskor.fh-aachen.de/activities/CO2Meter/
- [3] Makerspace des Fachbereichs Elektrotechnik und Informationstechnik der FH Aachen: http://makerspace.fh-aachen.de/
- [4] Offene Hardware- und Software-Dateien für das Projekt: https://makerspace-ac.de/seite/
- [5] "Datasheet Sensirion SCD30 Sensor Module", Version1, Mai 2020: https://bit.ly/3eXJa1V
- [6] Aktuelle Software-Revision auf GitHub: https://github.com/co2mtr/co2meter.git
- [7] Kontakt, um ein Kit zu erhalten: https://makerspace-ac.de/#contact-section
- [8] T. Dey, I. Elsen, A. Ferrein, T. Frauenrath, M. Reke and S. Schiffer:
- "CO<sub>2</sub> Meter: A do-it-yourself carbon dioxide measuring device for the classroom", PETRA 2021



## Nostalgisches MW/LW-Radio mit MK484

...macht immer Spaß zu bauen!

Von Gert Baars (Niederlande)

Der Bau eines Radioempfängers gehört zu den Elektronikprojekten mit der größten Tradition. Dieser Empfänger basiert auf dem ehrwürdigen AM-Detektor MK484 und auch der LM386 als Audioverstärker ist keineswegs eine Neuheit. Dennoch eignet sich das Projekt hervorragend zum Experimentieren mit dem Radioempfang im Mittelwellenband (MW), obwohl dieses Band zunehmend verrauscht und von Hörfunksendern verlassen wird.





Bild 1. Der Empfänger besteht aus drei HF-Stufen und einem Audioverstärker.

#### Wie es funktioniert

Im Schaltplan **(Bild 1**) ermöglichen drei HF-Stufen den Empfang und die Erkennung von AM-Funksignalen im MW-Band (normalerweise 500...1600 kHz).

In der ersten Stufe sorgt T1, ein Sperrschicht-FET vom Typ J310, für eine positive Rückkopplung im Antennenabstimmkreis. Die Rückkopplung begrenzt die Dämpfung der Spule, erhöht die Empfindlichkeit und verringert die Gesamtbandbreite des Empfängers. Die Antenne L1 besteht aus einem 120...200 mm langen Ferritstab mit einem Durchmesser von 10 mm in einer eng gewickelten Spule mit 55 Windungen aus 0,2 mm durchmessenden Kupferlackdraht. CV1 ist ein 500-pF-Abstimmkondensator mit Glimmer- oder Luftspalt, der einen dem MW-Band entsprechenden Frequenzbereich bietet. Der Abstimmbereich kann durch Schließen des Bandschalters S1 bis auf etwa 150 kHz erweitert werden.

Die zweite Stufe besteht lediglich aus einem BF494 (T2) zur Erhöhung der Empfängerempfindlichkeit und Verstärkung des MW-Signals für den Eingang der dritten Stufe an IC1, besagtem MK484. Dieses dreipolige IC im TO92-Gehäuse wurde ursprünglich 1972 (!) von Ferranti als ZN414 auf den Markt gebracht und enthält einen kompletten HF-Verstärker, einen Detektor und eine AGC-Schaltung, was ihn zu einer der beliebtesten Lösungen für den Bau einfacher AM-Empfänger machte und heute noch macht. Obwohl die ursprünglichen Mitglieder der ZN41x-Serie längst veraltet sind, gibt es moderne(re) Äquivalente wie den MK484.

Im Anschluss an die HF-Abteilung haben wir noch den guten alten LM386 (IC2), der als Audioverstärker und Lautsprechertreiber fungiert, komplett mit dem vorgeschriebenen Boucherot-Netzwerk C11/R10. Vielen Lesern dürfte der 386 ein Begriff sein.

Der Empfang ist optimal, wenn der Bandbreitenregler P1 richtig eingestellt ist - die Einstellung erfolgt einfach nach Gehör. Der Abstimmbereich kann mit einem Festkondensator C1, der über S1 in die Schaltung eingebunden wird, bis hinunter zur Langwelle erweitert werden. Durch die dadurch entstehende zusätzliche Dämpfung kann zwar die Mitkopplung bei tiefen Frequenzen übermäßig abfallen, was sich aber glücklicherweise durch einen kleineren Wert für den Kondensator C3 ausgleichen lässt. Möglicherweise sind einige Einstellungen, Optimierungen und Experimente erforderlich, um den richtigen Wert für MW und LW zu finden. Ein größerer Wert für C1 kann auch dazu beitragen, den Abstimmbereich zu erweitern. Bei Frequenzen oberhalb der offiziellen MW-Frequenz von etwa 1,6 MHz ist es unwahrscheinlich, dass Sie (legale) AM-Sender finden.

Durch die Montage von zwei Ferritstäben nebeneinander kann die billige und einfache Antenne sowohl in der Richtwirkung als auch in der Selektivität verbessert werden.

#### **Technische Daten**

Der Empfänger wird mit einer Gleichspannung von 6...12 V betrieben, so dass idealerweise eine 9-V-Batterie eingesetzt werden kann. Der typische Strombedarf liegt im Bereich von einigen Milliampere. Bei voller Audioleistung werden allerdings einige hundert Milliampere benötigt. In den Abendstunden ist bei guten Ausbreitungsbedingungen der Empfang im MW-Band optimal, und es können bis zu 1000 km entfernte Stationen nur mit der einfachen Ferritantenne gehört werden!

200372-03

#### Haben Sie Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an die Redaktion unter luc.lemmens@elektor.com oder kontaktieren Sie Elektor unter redaktion@elektor.de.

#### **Ein Beitrag von**

Idee und Entwurf: Gert Baars Text: Gert Baars, Luc Lemmens Illustrationen: Patrick Wielders Redaktion: Jan Buiting, Luc Lemmens Übersetzung: Rolf Gerstendorf Layout: Giel Dols



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 nachlayouten.
- 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

Anfragen/Bestellungen: Ipn@lp-nord.de Telefon 04531 1708 0 lekt

Wenn der Winter naht und die Nächte länger werden, zieht es die Menschen zum Licht. Und auch zu diesem Thema hatte Elektor in sechs Jahrzehnten einiges zu bieten. Wir haben ein Dutzend Artikel und Projekte herausgesucht: Es geht um die voranschreitende (LED-)Technik, um clevere Steuerungen, aber auch um die Stimmung - von Mensch und Pflanze.

lektor

lektor

lektor

ekto

≥kto

ektor




### LEP - Licht emittierende Pistole (1973)

Zum Thema Steuern mit Licht gab es diverse Schaltungen in Elektor. Eine davon ist diese aus den 70ern. So ein "Spielzeug" gab es auch zu kaufen, doch der Selbstbau machte natürlich mehr Spaß.

#### www.elektormagazine.de/magazine/elektor-197312/55393



### Alle Jahre wieder (Blinkstern) (1988)

Ein Weihnachtsprojekt darf in unserer Zusammenstellung nicht fehlen. Kernstück der Schaltung: Ein 14-stufiger Binärzähler, dessen Ausgänge eine Leuchtdiodenmatrix steuern.

www.elektormagazine.de/magazine/elektor-198812/48757



### Warnblinker mit Blitzröhre (1984)

Gedacht war das Projekt als "Notsignalgeber für Autofahrer, Freizeitkapitäne und Bergsteiger". Es zeigte, wie sich eine gasgefüllte Blitzröhre durch eine Autobatterie oder vier 1,5-V-Zellen ansteuern ließ.

www.elektormagazine.de/magazine/elektor-198406/47768



### Mehr Licht -High-Power-LEDs in der Praxis (2003)

High-Power-LEDs waren nun endlich auch für private Anwender erhältlich. In diesem Artikel wurden beleuchtungstechnische Grundlagen vermittelt, zum anderen zwei einfache Schaltungen vorgestellt, die den Betrieb von 1-W-LEDs ermöglichten.

www.elektormagazine.de/magazine/elektor-200310/1750

#### **Kostenloser Download!**







### LED-Bio-Lichttherapiegerät (2006)

Die recht simple Schaltung soll - eingebaut in das Gehäuse eines Nasenhaar-Rasierers - Heuschnupfen lindern. Dass es hilft, konnte Elektor nicht garantieren, den Spaß am Löten aber schon!

www.elektormagazine.de/magazine/elektor-200607/2484





### Wecken mit Licht (2011)

Der hier vorgestellte Lichtwecker imitiert einen Sonnenaufgang - für einen "besseren Start in den Tag". Die Zutaten dieses Projekts sind ein DCF77-Modul, ein ATmega168, ein 2x16-Zeichen-LCD-Modul und vier Taster.

www.elektormagazine.de/magazine/elektor-201102/3743

### Surround-Light (2008)

Ein "Sphärenlicht" für Computer-Monitore und TV-Geräte, bei der farbige LEDs rein analog anhand des Bildschirminhalts angesteuert werden, der über SCART zur Schaltung gelangt. Nicht ganz trivial, und gerade deshalb lesenswert!

www.elektormagazine.de/magazine/elektor-200802/2902

#### Kostenloser Download!



### Lumina (2015)

Die Bluetooth-LE-RGBW-Lampe war ein gelungene Mischung aus zeitgemäßer Hardware und cleverer Software - und lud wie immer nicht nur zum Nachbau (Bild), sondern auch zum Modifizieren und Erweitern ein.

www.elektormagazine.de/magazine/elektor-201512/28229

#### Kostenloser Download!

### Dimmer mit Berührungsschalter (2010) Dimmer gab es zuhauf in 60 Jahren Elektor. In dieser "Halbleiter-

heft"-Schaltung wurde ein IC von LSI Computer Systems verwendet, direkt versorgt aus dem Netz.

www.elektormagazine.de/magazine/elektor-201007/3539





### Bio-Light (2018) und Floranium (2018)

Pflanzen erzeugen elektrische Signale, und diese lassen sich mit farbigen LEDs sichtbar machen. Zu diesem Thema gab es gleich zwei Projekte in Elektor.

www.elektormagazine.de/magazine/elektor-201801/41210

www.elektormagazine.de/magazine/elektor-201805/41506



### Horticulture Box: LED-Licht für Pflanzen (2019)

Die pflanzenfreundliche LED-Beleuchtung soll das Wachstum stimulieren. Im Zeitalter des Internet of Things darf ein ESP32-Webinterface natürlich nicht fehlen!

www.elektormagazine.de/magazine/elektor-109/51129

210421-02

#### **Ein Beitrag von** Autor und Redaktion: Jens Nickel Layout: Harmen Heida



### **PASSENDE PRODUKTE**

> Elektor Artikel-Archiv auf USB-Stick (SKU 19197) www.elektor.de/19197

### Welches war euer schönstes Titelbild?

Joachim Wülbeck @4jochen · May 22 60 years @Elektor #elektor60 and 50 years @ElektorDE in Germany #elektor50 our oldest is from 1970 and all without a single missing since 1972 ;-)



I project that I ever built - the BASIC d on an 8052AH-BASIC. I still use it.

me DAT tape, tool



Halbleiterheft 1979 war das erste Heft. Waren das noch Zeiten als man 5%-Widerstände ernten konnte! 😭 60 Jahre Elektor, 42 davon für mich, wie die Zeit vergeht...

••••



den sozialen Netzwerken eine kleine Umfrage gestartet. Über LinkedIn und Twitter haben uns eine Vielzahl von Antworten erreicht, vielen Dank hierfür!

https://twitter.com/ElektorDE www.linkedin.com/company/elektor-international-media





 $\rightarrow$  C

o www.elektor.de

## **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.com). Unsere Bedingungen: **Nie teuer, immer überraschend!** 

### Elektor 2 MHz LCR-Meter Kit



### Preis: 799,00 € Mitgliederpreis: 719,10 €

ੇ ਯww.elektor.de∕19883

### Anycubic Photon Mono X – UV Resin SLA 3D-Drucker

Preis: 589,00 € Mitgliederpreis: 530,10 €

🖵 www.elektor.de/19831



Joy-Pi Note – 3-in-1 Lösung: Notebook, Lernplattform und Experimentierzentrale



Preis: 369,00 € Mitgliederpreis: 332,10 €

ੇਂਟ੍ਰੇ www.elektor.de/19877



Pokit Meter – Portables Multimeter, Oszilloskop und Logger

Preis: 104,95 € Mitgliederpreis: 94,46 €

 $\Delta$ 

ਊ www.elektor.de∕19854



Raspberry Pi 400 Buffer Board

Preis: 21,95 € Mitgliederpreis: 19,76 €

₩www.elektor.de/19884

DFRobot Bluno – Arduino-kompatibles Board mit Bluetooth 4.0



₩www.elektor.de/19878



# 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 o 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 o bis F (also o 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:

0

Elektor Redaktion Kackertstr. 10 52072 Aachen

#### Fax: 0241 / 955 09-013 E-Mail: hexadoku@elektor.de

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 17. Dezember 2021.

### **DIE GEWINNER DES HEXADOKUS AUS DER AUSGABE SEPTEMBER/OKTOBER STEHEN FEST!**

#### Die richtige Lösung ist: 0718D

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 bekannt gegeben.

			С	3							0	В			
	5				В	0	7	1	9	С				8	
В	F		8			4			2			5		0	9
0			7	8		9			5		4	6			Е
С	В					D			8					4	0
	А	0		Е			4	2			7		5	6	
		3	4	9	F	1			Е	0	5	D	8		
	2	8			3					1			F	9	
	Е	В			9					Α			0	3	
		5	D	4	С	Е			0	В	9	А	7		
	4	9		5			D	8			1		Е	В	
8	С					Α			7					1	D
2			6	7		F			С		В	0			3
5	3		В			8			1			7		F	2
	D				Α	2	В	3	F	6				5	
			F	1							2	8			

Herzlichen Glückwunsch!

Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.

# Treten Sie jetzt der Elektor Community bei!

GÓLD



Jetzt



Mitglied werden!

- ✔ Komplettes Webarchiv ab 1970
- ✓ 6x Elektor Doppelheft (Print)
- ✓ 9x Digital (PDF) inkl. Elektor Industry
- 10% Rabatt im Online-Shop und exklusive Angebote
- Elektor Jahrgangs-DVD

- Mit Tausenden von Mitgliedern des Online-Labors gemeinsam entwickeln mit Zugang zu über 1.000 Gerber-Dateien und direktem Kontakt zu unseren Experten!
- Veröffentlichen Sie Ihr eigenes Projekt oder verkaufen Sie direkt über unseren Shop!

### Auch erhältlich



- 🇹 Zugang zu unserem Webarchiv
- Y 10% Rabatt in unserem Online-Shop
- 🌱 6x Elektor Doppelheft (PDF)
- 🖌 Exklusive Angebote
- 🗹 Zugang zu über 1.000 Gerber-Dateien



### www.elektor.de/mitglied



## Paul Eisler

Paul Eisler hat die Leiterplatte erfunden und somit die Miniaturisierung von elektronischen Schaltungen maßgeblich vorangetrieben. Im Februar 1943 meldete Eisler das Patent 639,178 "Manufacture of Electric Circuits and Circuit Components" in London an, von wo es routinemäßig auch an das amerikanische Bureau of Standards gemeldet wurde.

Als in den frühen 1940er Jahren in den USA ein Annäherungszünder für Flugabwehrgeschosse entwickelt wird, griff man dort auf Eislers Erfindung zurück, und führte sie zur Serienreife. Damit begann der weltweite Siegeszug der Leiterplatte.

Paul Eisler ist ein Pionier. **Sei auch Du ein Pionier** und setze deine Idee noch heute um. Den Prototypen dazu fertigt **AISLER**.

### Schnell, günstig, ökologisch, direkt aus Europa.

### Platinen

Industriell gefertigte 2- und 4-Lagige Platinen ab zwei Arbeitstagen, z.B. 50mm x 50mm 2 Lagen für 11,20 € inkl. Mwst. und Versand.

### Bauteile

Das weltweit größte Bauteillager. Kostengünstig und direkt für dein Projekt abgepackt und das ohne Versandkosten und Mindesbestellwert.

### SMD-Schablonen

Laser-gefertigte SMD Schablonen schon ab 10,00 € inkl. Mwst. und Versand. Egal ob einoder doppelseitig und unabhängig der Pad-Anzahl.

### Bestückung

Keine Zeit für die Bestückung deines Projekts? Kein Problem, dein Projekt kann auch komplett bestückt von uns an dich geliefert werden.

Besuche jetzt **aisler.de** und erhalte 10 € Rabatt auf deine Bestellung mit dem Rabattcode **TPRYFGB** 







AISLER