

Sommer- Ausgabe

Mit **extra vielen** Schaltungen
für Einsteiger und Fortgeschrittene



Wie macht man (Schöne)
Fotos von Projekten? 44

viele Infos
für Maker!

Analoge
Elektronik!



Elektor Sommer-Präsente:
Mitmachen und Gewinnen! 18



Einfacher
Funktionsgenerator 20



Review: Labornetzteil-
Kit • Joy-iT RD6006 28



Mein Projekt:
NeoPixel-LED-Uhr 76

In dieser Ausgabe

- > Temperaturregler mit Raspi
- > Ultraschall-Waschmaschine
- > Booster für weiße LED
- > Kurzschlussverfolgung
- > Einstieg in Node-RED
- > Verzerrer-Pedal
- > DIY-WLAN-Antenne
- > Signalgenerator mit ATtiny13
- > Minimal-Oszilloskop
- > Einparkhilfe mit Arduino

und vieles mehr!



KI-Praxis für Einsteiger
Neuronale Netze mit Python

110



YouTuber GreatScott!
baut eine LoRa-Alarmanlage

24



Info-Display für den
Heimgebrauch Mit Windows
auf dem Raspberry Pi

8



50 Jahre Elektor

Seit **Mai 1970** gibt Elektor eine deutsche Ausgabe heraus. In diesem Jahr feiern wir **fünf Jahrzehnte** innovativer Elektronik-Projekte, informativer Ingenieur-Tutorials und Fachartikel zu einer Vielzahl von technischen Themen. Schließen Sie sich uns an, um weitere **50 Jahre Elektronik** zu entwerfen, zu teilen und zu verkaufen!

Elektor-Archiv 1970-2019 (USB-Stick)

BESTELLEN SIE NOCH HEUTE!

www.elektor.de/1970-2019



51. Jahrgang, Nr. 574
Juli/August 2020
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

Margriet Debeij (verantwortlich)
Tel. 0241 95509174
Mobil: +49 170 5505396
E-Mail: margriet.debeij@elektor.com

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

Es gilt die Anzeigenpreisliste ab 01.01.2020.

Distribution

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

Druck

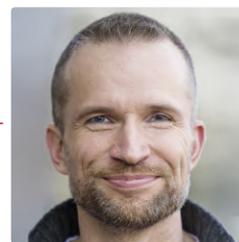
Pijper Media, Groningen (NL)

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.

© 2020 elektor international media b.v.

von Jens Nickel

Chefredakteur Elektor



Extra viele Schaltungen!

Die Corona-Krise hat uns alle fest im Griff, und vermutlich wird das noch lange so bleiben. Während ich diese Zeilen schreibe, sieht man aber zumindest wieder einen Silberstreif am Horizont - es gibt erste Lockerungen der Kontaktbeschränkungen und auch aus der Wirtschaft (darunter der Elektronik-Branche) kommen wieder gute Nachrichten. Wir alle hoffen natürlich, dass es weiter aufwärts geht und auch der *electronica Fast Forward Award powered by Elektor* auf der größten Elektronikmesse der Welt im November wie geplant stattfinden kann - im Moment sieht es ganz danach aus, worüber wir uns zusammen mit den Partnerfirmen und den beteiligten Start-ups (wir haben schon 40 Bewerber) sehr freuen.

Dennoch sind die Monate, in der wir mit einer großen Ungewissheit leben mussten, auch an uns nicht spurlos vorbei gegangen. Den Laborbetrieb haben wir zum Beispiel zurückfahren müssen, weil die Gesundheit der Kollegen selbstverständlich an erster Stelle stand. Auch mussten wir unsere Kosten generell im Blick behalten - nicht zuletzt dank eines stark zurückgegangenen Einzelheftverkaufs und vorsichtigen Anzeigenkunden, die ihr Marketingbudget für die Krisenmonate auf 0 gesetzt haben (was wir ihnen wirklich nicht übel nehmen können). Wir haben uns daher frühzeitig entschlossen, das Konzept dieser Ausgabe ein wenig anzupassen. Meine Kollegen und ich haben die letzten Wochen aber hart daran gearbeitet, dass diese Ausgabe mitnichten ein „Spar-Blatt“ wird. Passend zum Sommer - und in der langen Tradition der *Halbleiterhefte* - präsentieren wir Ihnen hier ein Heft voller kleiner Projekte und Schaltungen, die sich für Anfänger, aber auch Fortgeschrittene eignen. Dabei sind viele rein analoge Schaltungen, bei denen Sie ohne Computer und Programmiercode auskommen; doch auch die Freunde von Raspi, Arduino & Co. kommen auf ihre Kosten. Und wir haben noch ein Bonbon: Es gibt wertvolle Tools für Entwickler zu gewinnen (siehe Seite 18).

Ich brauche hier kaum noch einmal zu erwähnen, dass die Krise auch Chancen bietet - sicher folgen Sie uns auch online auf *elektormagazine.de*. Dort haben wir vielfach über Initiativen aus der Elektronikwelt berichtet, die Menschen in den besonders betroffenen Ländern helfen. Als international wohl wichtigstes Elektronikmagazin für Profis und Hobbyisten sind wir uns bewusst, dass wir bei der Vermittlung von Wissen und Informationen hier eine Schlüsselstellung besetzen. Über unsere eigene Initiative *Elektor Helps!* lesen Sie mehr auf Seite 6.

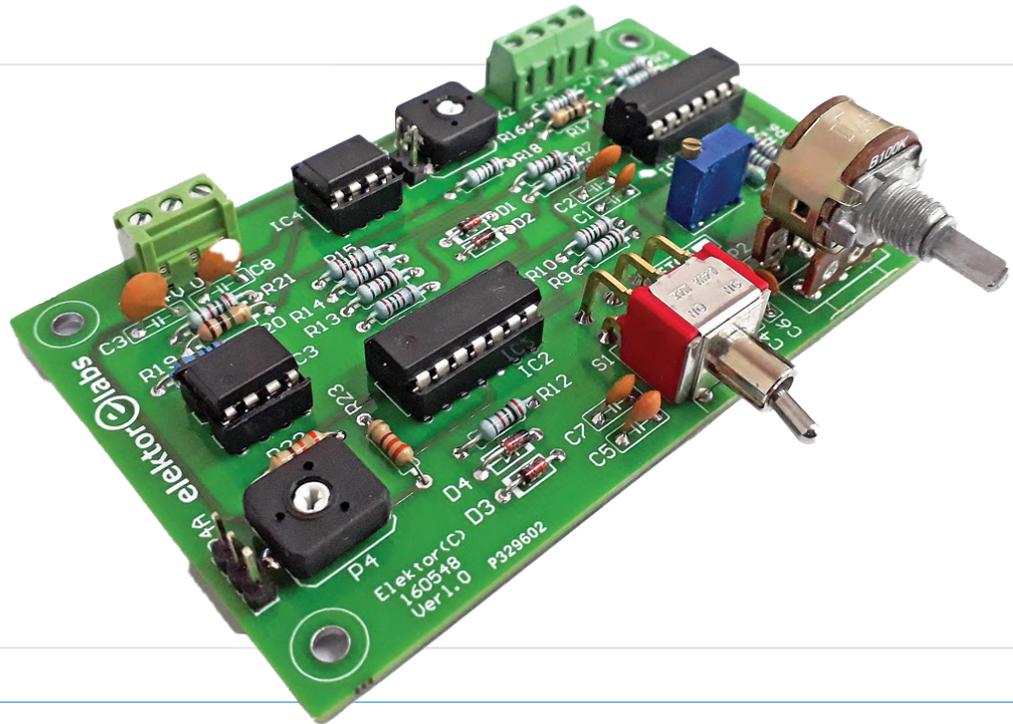
Bleiben Sie mit uns dran, und bleiben Sie gesund!

Unser Team



Chefredakteur: Jens Nickel (v.i.S.d.P.) (redaktion@elektor.de)
Redaktion: Eric Bogers, Jan Buiting, Rolf Gerstendorf, Denis Meyer, Dr. Thomas Scherer, Clemens Valens
Leserservice: Ralf Schmiedel
Elektor-Labor: Mathias Claußen, Ton Giesberts, Hedwig Hennekens, Luc Lemmens, Clemens Valens, Jan Visser
Grafik & Layout: Giel Dols
Herausgeber: Don Akkermans

Labor-Projekt Einfacher Funktions- generator 20



Rubriken

- 3 Impressum**
- 6 Elektor Ethics**
Elektronik in herausfordernden Zeiten - Elektor hilft!
- 36 Kleine Schaltungen - XXL**
- 70 Elektor Labs Pipeline**
- 74 Mein Labor**
Brauchen Sie wirklich dieses ganze Zeug?
- 76 Mein Projekt**
Uhr mit NeoPixel-Ringen
- 107 Hexadoku**
Sudoku für Elektroniker

- 68 Review: Elektor SDR-Praxis-Bundle**
- 72 Elektor „Kickstarter“**
Neue Funktion auf Elektor Labs!
- 78 Hello, World! Wir sind Elektor, wir sind sozial!**
- 80 M4 + 2x A7 + GPU: Ein ungleiches Dreamteam**
Der neue STM32MP1-SoC für gehobene Ansprüche
- 96 Preiswerter E-Scooter**
Was taugt ein „zugelassener“ 300-€-E-Tretroller von Lidl?
- 104 Grundlegendes für den Elektronik-Arbeitsplatz**
- 110 KI für Einsteiger (2)**
Neuronale Netze mit Linux und Python

Hintergrund

- 15 Einstieg in Node-RED**
Ein visuelles blockbasiertes Open-Source-Programmierwerkzeug
- 18 Elektor Sommer-Präsente**
- 28 Review: Labornetzteil-Kit Joy-iT RD6006**
- 33 Review: Interface-Board GreatFET One**
- 44 Wie macht man (schöne) Fotos von Platinen ...**
... und elektronischen Bauteilen?
- 48 Review: I²C-Driver Core**
- 51 Review: Touchscreen Joy-View 13**
- 58 Review: Signalgenerator Joy-iT JDS2915**

Projekte

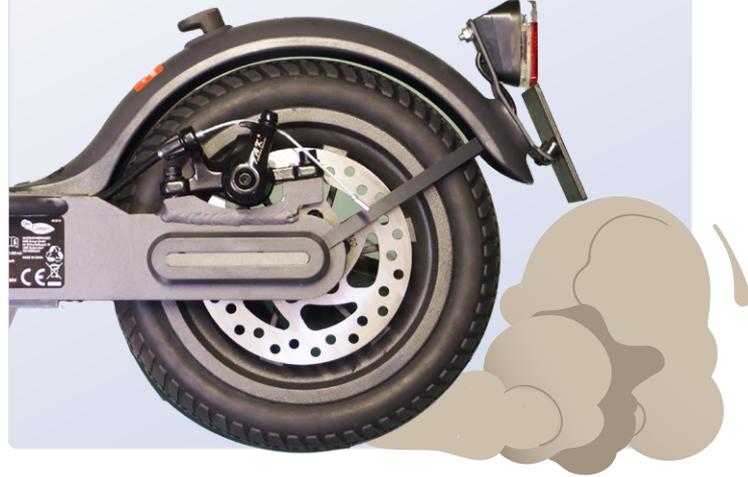
- 8 Info-Display für den Heimgebrauch**
Mit Windows auf dem Raspberry Pi
- 20 Einfacher Funktionsgenerator**
Mit umgekehrter Signalerzeugung
- 24 GreatScott! baut ein LoRa-Alarmsystem**
- 41 Des armen Mannes externe 2,4-GHz-WLAN-Antenne**
- 42 Einfacher Ein-Aus-Temperaturregler mit Raspberry-Pi-HAT**
- 54 LED-Booster für Mikrocontroller**
Mit nur einem Bauteil!
- 57 Experimentelle Ultraschall-Waschmaschine**
- 62 Ampelsteuerung in PIC-Assembler**
- 65 Der ewige Blinker**

M4 + A7 + GPU: Ein ungleiches Dreamteam



Preiswerte E-Scooter

96



Elektor Helps

Elektronik in
schwierigen Zeiten

6

- 66 Experimenteller Hall-Sensor
- 67 Kurzschlussverfolgung mit dem Milliohm- oder ESR-Meter
- 87 Testing Rig für 16F18877 und ähnliche PICs
- 88 Mikrocontroller-Kommunikation über SPI
- 90 Sechs Oszillatoren (und der Millereffekt)
- 92 Absolutes Minimal-Oszilloskop mit LED-Display
- 93 Knight Rider: LED-Lauflicht mit dem ESP32
- 94 LW/MW-AM-Signalgenerator mit ATtiny13
- 95 Minimalistisches Dipmeter
- 100 Ultraschall-Rückwärtseinparkhilfe mit Arduino Uno
- 102 Verzerrer-Pedal mit Operationsverstärker und Raumladungsröhren

Vorschau

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

Aus dem Inhalt:

- > Nixie-Bargraph-Thermometer Update
- > HV-Stromversorgung mit Kennlinienschreiber
- > Hausautomatisierung mit Home Assistant
- > KI für Einsteiger: Eigenes Neuronales Netz entwickeln
- > Open-Network-Wetterstation: Software
- > Röhren-Signalgenerator
- > FFT auf dem Maixduino
- > PICs programmieren mit Assembler
- > NeoPixel Jewel mit ESP8266 steuern

Und vieles mehr!

Änderungen vorbehalten.

Elektor September/Oktober 2020 erscheint am 3. September 2020.

Besuchen Sie die
neue Labs-Website!



e lektorhilft

Elektronik in schwierigen Zeiten

Kleiner Wettbewerb für „Lockdown-Projekte“

Freie Mitgliedschaften für diejenigen, die es am dringendsten brauchen



Elektor-Industry-Magazine hilft

- Ersatz von Veranstaltungen
- Vier monatliche Ausgaben mit redaktionellen Webinaren und Podcasts

offenes Elektor-Netzwerk

offene Elektor-Plattform

Von Erik Jansen

Die Corona-Krise hat die Welt erfasst. Natürlich ist es zunächst eine Katastrophe, wie sie noch keiner von uns je erlebt hat. Persönliche Tragödien, hart getroffene Länder und globale Auswirkungen auf die Wirtschaft und viele Branchen. Auch die Elektronikindustrie kann sich einer nachhaltigen Beeinträchtigung nicht entziehen.

Auch Elektor ist in verschiedenen Bereichen betroffen, und es erfordert einen wachen und kreativen Ansatz, um diese Zeit gemeinsam und ohne ernsthaften Schaden zu überstehen. Unser gesamtes Team ist bemüht, dies so gut und mit so wenigen Rückschlägen wie möglich für unsere Mitglieder und Kunden zu bewältigen. Aber wir haben die Hoffnung, dass wir Erfolg haben werden!

Zeiten wie diese zeigen uns auch positive Dinge. Ein starkes Gefühl der Solidarität und die vielen Initiativen, die daraus entstehen, sind fast überall zu sehen. Große und kleine herzerwärmende Projekte tragen dazu bei, die Welt ein wenig besser zu machen. Und trotz der Tatsache, dass wir uns bei Elektor auf unsere eigenen Belange konzentrieren müssen, ist Elektor daran interessiert, als Team und als Unternehmen einen Beitrag zu leisten. Auch wenn wir in manchen Fällen wenig selbst einbringen können, möchten wir den Initiativen, die wir um uns herum sehen, doch unsere Hilfe anbieten.

Unter dem Motto „**Elektor Helps**“ öffnen wir deshalb unsere Plattformen für alle, die unsere Werkzeuge, unsere Möglichkeiten und unser Wissen logisch und sinnvoll für eine gute Sache einsetzen können. Und zwar kostenlos.

Elektor Helps: Inhalte

Als Unternehmen haben wir festgestellt, dass es in diesen Zeiten nicht selbstverständlich ist, dass unsere Inhalte und Dienstleistungen leicht erhältlich sind. Aufgrund verschiedener Sperrungen, begrenzter Postdienste und natürlich neuer und anderer Prioritäten hat derzeit nicht jeder Ingenieur Zugang zu unseren Inhalten. Auch wenn immer mehr Länder jetzt einige der größten Einschränkungen aufheben, sind wir der Meinung, dass wir immer noch zu einer gewissen Erleichterung beitragen können. Aus diesem Grund bieten wir unter dem Label **Elektor Helps** Ingenieuren und Herstellern in den am stärksten betroffenen Ländern passende Lösungen an, die von **reduzierten Versandkosten bis hin zu einer kostenlosen Vollmitgliedschaft bei Elektor Green** für diejenigen reichen, die unter den schwierigen Umständen leiden. Auch für unsere Partner aus der Industrie ist es jetzt eine schwierige Zeit, da wichtige Veranstaltungen abgesagt sind und es schwierig ist, in Bezug auf Marketing und Kommunikation im Voraus zu planen. Auch in diesem Bereich wollen wir helfen. Es hat jetzt nicht viel Sinn, eine Sonderausgabe in Verbindung mit einer Veranstaltung zu planen. Es ist jedoch sinnvoll, diesen Veranstaltungen und ihren Besuchern eine Plattform zu bieten, um die Botschaft, die dort hätte übermittelt werden sollen, auf alternative Weise zu konkretisieren.

Die kommenden Ausgaben unseres Elektor Industry Magazine werden daher häufiger und multimedialer erscheinen. Ein monatliches Thema, das sich an den aktuellen Bedürfnissen und der aktuellen Situation orientiert, wird online, als E-Publikation, aber auch in Form von redaktionellen Webinaren und Podcasts ausgestaltet. Darüber hinaus stellen wir all diese Inhalte unserer wachsenden Community frei zur Verfügung. Die elektronische Sonderausgabe *Elektor hilft - Elektronik in schwierigen Zeiten / Elektor Helps - Electronics in challenging times* wird in den nächsten vier Monaten an mindestens 120.000 Ingenieure weltweit verschickt werden.

Elektor Helps: Projekte

Elektor unterstützt seit Jahren Ingenieure bei Projekten auf unserer Online-Plattform Elektor Labs. Es ist ein Ort, an dem eine wunderbare Vielfalt von Projekten zusammenkommt. Dies bildet den Kern unserer Existenz, und daran werden wir natürlich nichts ändern.

Wir haben das Gefühl, dass es auch im Elektor-Labor viele Menschen gibt, die seit den Schließungen ihre Fähigkeiten in die Arbeit eingebracht haben, was zu großartigen neuen Ideen geführt hat, die für andere hilfreich sein können. Ob zum Spaß oder für die Gesellschaft, wir möchten unsere Community ermutigen, diese Projekte auf der Elektor-Labs-Plattform zu teilen. Das bedeutet nicht, dass es sich dabei um ein Beatmungsgerät handeln muss. Es kann auch ein nettes kleines Projekt sein, das Spaß macht und das man mit einer großen Gruppe von Gleichgesinnten teilen kann, von denen einige derzeit gegen ihren Willen zu Hause sind und sich langweilen. Etwas, das Spaß macht, kann auch helfen!

Wir sind uns natürlich sehr wohl bewusst, dass es bereits viele Initiativen gibt, die Ingenieure dazu aufrufen, ihr Wissen und ihre Fähigkeiten für gute Zwecke einzusetzen. Dennoch möchte Elektor hier einen kleinen Anstoß geben. Nicht, um anderen guten Initiativen das Rampenlicht zu nehmen, sondern vielmehr, um ihnen mehr Aufmerksamkeit zu schenken oder das letzte bisschen Motivation zu geben, das eine Projekt, das bereits auf Elektor Labs herangereift ist, noch einmal zu überdenken. Dort können Sie sich auch unseren kleinen Wettbewerb ansehen!

Elektor Helps: Netzwerk

Als Unternehmen, das seit sechzig Jahren aktiv ist (in diesem Jahr feiert Elektor sein fünfzigjähriges Jubiläum in Deutschland!), hat Elektor natürlich ein großartiges Netzwerk aufgebaut. Und das nicht nur mit rein geschäftlichen Verbindungen! Mehr als einmal hörten wir schon von Spitzenleuten in sehr großen Unternehmen, dass ihre Leidenschaft für Elektronik mit der Lektüre von Elektor begann. Einige unserer Partnerfirmen haben bereits angedeutet, dass sie ausgewählten Initiativen im Rahmen des Projekts **Elektor Helps** in geeigneter Weise Hilfe anbieten möchten, von Produktionskapazität bis zu Support. Wir sind nicht allein!

Jenseits von Corona

Vor einigen Jahren war Elektor unter den ersten, die das Thema „Ethik in der Elektronik“ in die Diskussion eingebracht hat. Und etwas früher hatten wir unsere geballte Denkkraft für die niederländische Asthmastiftung eingesetzt, indem wir mit ihr ein CO₂-Messgerät entwickelt haben. Diese neue Initiative **Elektor Helps** passt gut dazu. Sie können also von uns erwarten, dass wir, wenn auch wahrscheinlich in einem anderen Format, weiterhin große und kleine Projekte unterstützen werden. Von Schulen, die vielleicht ein Problem haben, bis hin zum Vervollständigen von Kreisläufen für eine bessere Welt: **Elektor hilft!** ◀

200194-04

[1] www.elektormagazine.com/helps-content





Info-Display

für den Heimgebrauch

Mit Windows auf dem Raspberry Pi

Von **Dr. Veikko Krypczyk**

Informationssysteme zeigen das Wetter, aktuelle Nachrichten, die Abfahrtszeiten der naheliegenden Stadtbahnlinie oder andere aktuelle Daten an. Man kann solche Infosysteme bestens im Heim- und Unternehmensbereich einsetzen. Eine geschickte Auswahl aus Hard- und Software macht das System flexibel.

Informationssysteme sind en vogue und können für die unterschiedlichsten Zwecke im privaten und betrieblichen Umfeld eingesetzt werden. Ein privates Home Information System oder kurz *Home-InfoSystem* könnte den Nutzer beispielsweise mit folgenden Informationen versorgen:

- Das aktuelle Wetter des Standorts und gegebenenfalls eine Vorausschau für die kommenden Tage
- Nachrichten, lokal, landesweit oder weltweit, gegebenenfalls eingeschränkt auf einen bestimmten Themenbereich wie Sport oder Kultur

- Fotos, präsentiert in Form einer Diashow, aus der Cloud
- das abendliche TV- oder Kinoprogramm
- die Abfahrtszeiten der naheliegenden Stadtbahn mit aktuellen eventuellen Verspätungsinformationen
- der aktuelle Ort der Familienmitglieder, sofern eine Geolocation-Funktion bei mobilen Geräten angebunden ist
- aktuelle oder nächste Termine aus dem Kalender.

Für den Unternehmensbereich (*EnterpriseInfoSystem*) gibt es auch eine Reihe von vom Einsatzzweck abhängigen Anwendungen. Auch dazu einige Beispiele:

- Anzeige von Arbeits- und Terminplänen, automatisch gespeist aus dem Teamkalender in der Cloud
- Soll- und Ist-Werte aus einem laufenden Produktionsbetrieb
- aktuelle betriebsinterne Informationen
- Informationen für Kunden, Teilnehmer von Veranstaltungen und so weiter.

Die Liste lässt sich beliebig fortsetzen und man findet mit Sicherheit noch viele interessante Anwendungen. Die Beispiele waren bisher nur auf eine passive Darstellung der Informationen ausgerichtet, aber es könnten auch interaktive Funktionen eingebunden werden, wenn man zum Beispiel einen Touchscreen verwendet.

Solche Informationssysteme kann man fertig kaufen oder in Eigenregie herstellen. Dabei kommt es auf das passende Zusammenspiel von Hard- und Software an. Beim Selbstbau bleibt man flexibel; man kann das System an die eigenen Bedürfnisse anpassen und ist nicht abhängig von der Software eines Anbieters. Dieser Artikel stellt den Ansatz für ein leicht zu realisierendes und nach eigenem Gusto modifizierbares HomeInfoSystems (HIS) vor. Die Software wird zur Verwendung und freien Weiterentwicklung auf [1] zur Verfügung gestellt. Das Prinzip wird mit dem Einsatz von Webdiensten (Wetter, News, Fotos) erläutert, so dass man das HomeInfoSystem nach seinen eigenen Bedürfnissen erweitern kann.

Systemaufbau

Das System soll für 24/7-Betrieb ausgelegt sein, so dass der Energieverbrauch – unabhängig von der gewählten Anzeige (Bildschirmgröße) – möglichst gering sein sollte. Aus diesem Grund bietet sich der Einsatz eines Mini-PCs in Form eines Einplatinen-Computers an, zum Beispiel ein Raspberry Pi, ein Rock Pi 4, ein Asus Tinkerboard S oder ein Banana Pi M3. Für den Raspberry Pi spricht der Preis, seine ausreichende Leistung, eine gute Unterstützung für die vorgesehene Aufgabe und vor allem umfassende Erfahrungen mit diesem Minicomputer. **Bild 1** zeigt den Aufbau des Systems. Im Mittelpunkt steht der Raspberry Pi. Das Modell 3B ist am besten geeignet, aber alternativ kann auch das ältere Modell 2B eingesetzt werden, da als Betriebssystem Windows 10 IoT verwendet wird. Das neueste Modell 4 bietet zwar deutlich mehr Leistung, wird aber nicht offiziell von Windows 10 IoT unterstützt.

In der Blockschaltung in Bild 1 ist zu sehen, dass der Raspberry Pi wie üblich von einem guten, stabilisierten Steckernetzteil mit 5 V und echten (!) 2 A versorgt wird. Die Netzwerkanbindung kann kabelgebunden über LAN oder drahtlos über WLAN erfolgen. Bei der Anzeige haben wir eine breite Auswahl, was Größe, Auflösung und die Verfügbarkeit eines Touch-Interfaces betrifft. Die Bildschirmgröße ist natürlich vom Anwendungszweck abhängig, die Palette reicht vom kleinen Bildschirm auf dem Tisch bis zur „Anzeigetafel“, die man auch aus größerer Entfernung ablesen kann. Der Raspberry Pi (Modell 2 und 3) schafft maximal eine Auflösung von 1.920 x 1.080 Pixel (Full HD). Eine höhere Auflösung bietet eine schärfere Darstellung und mehr Inhalte können platziert werden, aber es ist natürlich eine Kostenfrage, für welche Auflösung man sich entscheidet, vor allem, wenn ein Touchscreen gekauft werden soll. Ob Touch oder nicht, hängt von verschiedenen Kriterien ab:

- Kann/soll der Nutzer/Betrachter etwas auswählen?
- Ist eine Interaktion geplant?
- Ist der Bildschirm überhaupt in Reichweite des Nutzers?

Die Verbindung zwischen Raspberry Pi und Monitor erfolgt in jedem Fall über ein HDMI-Kabel. Wird ein Touchscreen verwendet, dann gibt

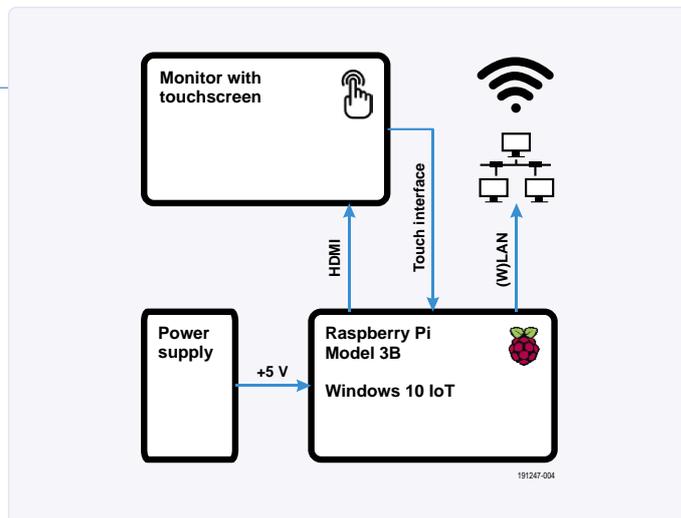


Bild 1. Systemaufbau des HomeInfoSystems (HIS).

es eine zusätzliche Verbindung zwischen Monitor und Raspberry Pi. Beim Raspberry Pi wird dabei ein USB-Port belegt, die Anschlüsse am Monitor aber unterscheiden sich. Neuere Modelle setzen meist auf einen USB-C-Anschluss.

Der Monitor muss selbstverständlich auch mit Strom versorgt werden. Ich verwende für mein HomeInfoSystem einen mobilen Full-HD-Touchscreen mit einer Diagonalen von 13 Zoll mit den Anschlüssen HDMI, USB-C für Touch, USB-C für die Stromversorgung und einem integrierten Akku. Eine Option ist es, den Raspberry Pi „huckepack“ an einem größeren Monitor zu montieren und diesen dann zum Beispiel an der Wand oder Decke zu befestigen. Das HDMI-Kabel kann dann besonders kurz ausfallen.

Architektur der Software

Als Betriebssystem auf dem Raspberry Pi wird Windows 10 IoT installiert. Für dieses Betriebssystem sprechen mit Blick auf ein Info-System die folgenden Eigenschaften:

- aktiver und direkter Kiosk-Modus, das heißt, eine App wird direkt ausgeführt, bei der der Nutzer nur eingeschränkte Zugriffsrechte besitzt.
- umfassende Optionen zur Gestaltung des Benutzerinterfaces in der Beschreibungssprache XAML
- leichte Administration des Raspberry Pi über das grafische Webinterface
- Entwicklung, Test- und Bereitstellung direkt aus der integrierten Entwicklungsumgebung Visual Studio
- automatisiertes Update der App über den Store möglich (Szenario für kommerzielle Verwendung).

Auf dem Raspberry Pi installiert man zunächst Windows 10 IoT (siehe **Kasten**). Für die Entwicklung wird ein PC (Desktop/Notebook) mit einer aktuellen Windows-10-Version benötigt, auf dem die App erstellt und auch getestet wird. Als Entwicklungsumgebung kommt Visual Studio zum Einsatz.

Es können nur die geräteübergreifenden Apps für die Laufzeitumgebung Universal Windows Platform (UWP) auf Windows 10 IoT ausgeführt werden, klassische Windows-Anwendungen laufen nicht. Apps für die UWP basieren auf .NET Core, werden in C# beziehungsweise in Visual Basic .Net und mit Hilfe der Entwicklungsumgebung Visual Studio erstellt. Es genügt die kostenfreie Community Edition [2]. Im Installer des Visual Studios wählt man das entsprechende Workload

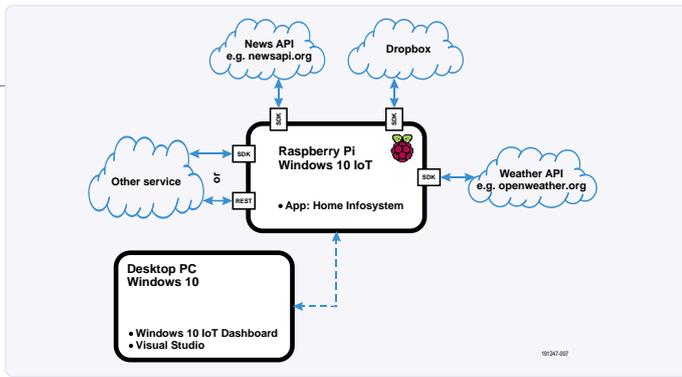


Bild 2. Die Software HomeInfoSystem im System.

für die Entwicklung von Apps für die UWP aus. Dann kann man mit der Entwicklung der ersten App beginnen. Dabei muss man beachten, dass man den Typ der Applikation von X86 auf ARM abändert, denn im Raspberry Pi arbeitet ein ARM-Prozessor.

Die Apps werden auf dem Entwicklungsrechner vollständig getestet, und erst, wenn eine Version fehlerfrei läuft, erstellt man das so genannte App-Package und installiert dieses über das Webinterface auf dem Raspberry Pi. Während der Entwicklung besteht auch die Möglichkeit, die App zum Testen und Debuggen direkt auf dem Raspberry Pi auszuführen. Voraussetzung für dieses Remote-Debugging ist nur, dass sich beide Rechner im gleichen lokalen Netzwerk befinden. Da Apps für die UWP auf .NET Core basieren, steht die gesamte Leistungsfähigkeit der zugehörigen APIs zur Verfügung. Für viele Standardprobleme kann man auf fertige Klassen zurückgreifen. Als Programmiersprachen kann man zwischen C# und Visual Basic .NET wählen.

Eine App wird auf dem Raspberry Pi im Vordergrund und im Vollbildmodus (Kiosk-Modus) ausgeführt. Das Userinterface wird in der Beschreibungssprache XAML deklarativ gestaltet. Dazu werden vordefinierte Steuerelemente in Layout-Container eingefügt. Auf diese Weise entsteht ein relatives Layout, das sich automatisch an unterschiedliche Bildschirmgrößen und -auflösungen anpasst. Die Palette der verfügbaren Steuerelemente umfasst die üblichen Elemente wie Buttons oder Textfelder. Starten Sie ein neues Projekt, gestalten Sie ein minimales Userinterface, führen Sie die App testweise auf Ihrem PC aus und übertragen Sie dann das App-Package auf den Raspberry Pi. Damit hat man den Entwicklungszyklus durchlaufen.

Kernelement des Softwaresystems ist die App *HomeInfoSystem* (Bild 2), die mittels App-Package und über das Webinterface auf dem Raspberry Pi installiert wird. Diese App dient primär der Präsentation der Inhalte, der Daten, die von den externen Services abgerufen werden. Ausnahme bilden statische Funktionen und Inhalte wie die Anzeige der Uhrzeit. Über ein Menü am linken Rand der App kann man auswählen, welcher Inhalt gerade angezeigt wird. Zwischen den Inhalten wird in einem wählbaren Zeitintervall automatisch gewechselt. Im aktiven Präsentationsmodus wird das Menü vollständig ausgeblendet. Lediglich das so genannte Hamburger-Menü-Icon  am oberen Rand erinnert daran, dass man es aufklappen kann. Der automatische Wechsel zwischen den Inhalten kann für jeden Service individuell konfiguriert werden. Die Anzeige der Fotos aus der Dropbox sollte etwas länger dauern (zum Beispiel 45 s), während für das aktuelle Wetter durchaus 20 s ausreichen dürften. Inhalte und Services können über drei Arten in die App integriert werden.

► **Intern:** Die Funktionalität bezieht keine weiteren Daten, wie zum Beispiel die Anzeige der aktuellen Uhrzeit. Die gesamte Funktion wird inklusive der Daten vollständig statisch in den Quellcode der App integriert.

► **Extern (SDK):** Für viele Services kann man Dienste von Anbietern nutzen. Dazu gehören zum Beispiel aktuelle Wetterdaten, News, der Zugriff auf Cloudspeicher-Dienste wie Dropbox sowie eine Kalenderintegration. Für viele dieser Dienste stehen so genannte Software Development Kits (SDK) zur Verfügung. Ein SDK ist speziell auf die Programmierumgebung, in diesem Fall auf .NET Core und C# zugeschnitten. Gegenüber einer generischen Programmierschnittstelle, zum Beispiel auf der Basis von REST, ergibt sich der Vorteil, dass die Einbindung und Programmierung stark vereinfacht wird. Ein solches SDK wird normalerweise

Listing 1. Wetterdaten (Beispiel) von <https://openweathermap.org>.

```

{"coord": { "lon": 139,"lat": 35},
  "weather": [
  {
    "id": 800,
    "main": "Clear",
    "description": "clear sky",
    "icon": "01n"
  }
],
  "base": "stations",
  "main": {
    "temp": 281.52,
    "feels_like": 278.99,
    "temp_min": 280.15,
    "temp_max": 283.71,
    "pressure": 1016,
    "humidity": 93
  },
  "wind": {
    "speed": 0.47,
    "deg": 107.538
  },
  "clouds": {
    "all": 2
  },
  "dt": 1560350192,
  "sys": {
    "type": 3,
    "id": 2019346,
    "message": 0.0065,
    "country": "JP",
    "sunrise": 1560281377,
    "sunset": 1560333478
  },
  "timezone": 32400,
  "id": 1851632,
  "name": "Shuzenji",
  "cod": 200
}
...

```

WINDOWS 10 IOT AUF DEM RASPBERRY PI EINRICHTEN

Die Installation des Betriebssystems auf die SD-Karte des Raspberry Pi erfolgt (genauso wie die Entwicklung der Software) von einem Windows-10-PC aus. Aktualisieren Sie gegebenenfalls im Vorfeld die Version von Windows 10. Windows 10 IoT Core liegt aktuell in der Version 1809 (Build 17763) vor. Ihr PC muss also mindestens auf diesem Stand sein. Die SD-Karte wird am PC in einen passenden (internen oder externen) Steckplatz gesteckt.

- › **Dashboard:** Ist alles bereit, lädt man die Setup-Datei für das Dashboard von Windows 10 IoT von [4] herunter, die die komplette Anwendung nachlädt.
- › **SD-Karte schreiben:** Im Dashboard (**Bild A**) wählt man den Menüpunkt *Setup a new device* und dort den richtigen Gerätetyp (Raspberry Pi 2 oder 3) sowie das Ziellaufwerk, die SD-Karte. Hilfreich ist, dass man hier direkt das aktuelle WLAN-Profil, Namen und Passwort für den Raspberry Pi auf die SD-Karte schreiben kann. Nach dem Akzeptieren der Lizenzbedingungen startet der Download und das Dashboard schreibt das Betriebssystem auf die SD-Karte.
- › **Start des Raspberry Pi:** Die beschriebene SD-Karte wird in den Raspberry Pi eingelegt und die vorgesehene Hardware (Monitor, Touch, Steckernetzteil) sowie hilfsweise Maus, Tastatur und eventuell ein LAN-Kabel angeschlossen. Der Raspberry Pi verfügt über keinen Ein/Ausschalter und startet

sofort, wenn er mit Strom versorgt wird. Die Ersteinrichtung dauert etwas länger. Nach der Auswahl der Sprache erscheint nach einigen Augenblicken der Startbildschirm von Windows 10 IoT. Es läuft noch keine benutzerdefinierte App.
› Die **Konfiguration des Raspberry Pi** erfolgt vom PC aus über das Dashboard beziehungsweise das Windows-Geräte-Portal (WDP) (**Bild B**). Beim Raspberry Pi Modell 3 ist WLAN direkt an Board. Windows 10 IoT erkennt die Hardware automatisch, so dass die Einrichtung problemlos verläuft. Beim Modell 2 ohne WLAN muss man einen externen WLAN-Dongle einstecken und einrichten, was etwas umständlicher ist als beim Modell 3. Für die Version 3 des Raspberry Pi spricht auch die höhere Rechenleistung.

Der Zugriff auf den Raspberry Pi mit Windows 10 IoT ist über einen Browser möglich. Das Dashboard zur Konfiguration erreicht man über die IP-Adresse des Raspberry Pi. Hier kann man unter anderem eine App installieren, starten und beenden, Statusinformationen zur Systemauslastung einsehen, Netzwerkverbindungen verwalten oder das System herunterfahren und neu starten. Alternativ kann man auch die Power-Shell (Kommandozeile) nutzen. Diese Art der Konfiguration ermöglicht die meisten Einstellmöglichkeiten, ist aber nicht sehr komfortabel. Sollen zusätzliche Hardwaretreiber eingerichtet werden, ist dies jedoch der einzige Weg. Wir erledigen hier alles über das Webinterface.

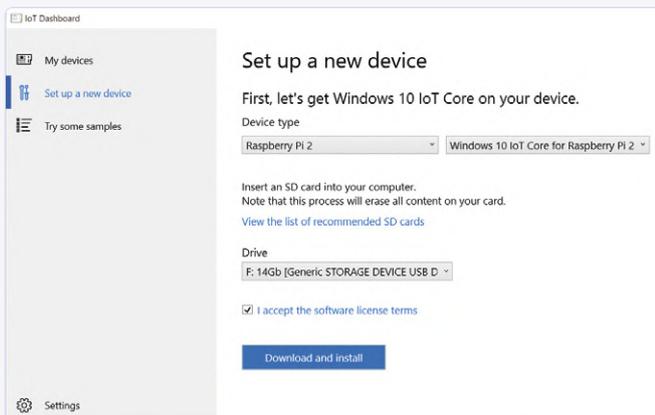


Bild A. Dashboard von Windows 10 IoT.

Nutzung und Weiterentwicklung des Quellcodes

Die Software zum Projekt auf [1] kann ohne Probleme direkt genutzt werden. Sie enthält: Startseite (Anzeige eines Begrüßungstextes), Uhr, Fotoshow mit Dropbox-Integration, Anzeige der Wetterdaten und Anzeige der News. Die Services sind vor der Nutzung minimal zu konfigurieren:

- › **Dropbox:** Im eigenen Dropbox-Account zum Entwickler-Bereich (Console) wechseln und eine App anlegen. Zugriff auf das Verzeichnis der App erlauben und App-Secret (Key) generieren. Dieser Key wird später in den Einstellungen der App eingegeben.
- › **News:** Melden Sie sich bei [5] an und generieren Sie kostenfrei einen persönlichen Key für den Zugriff auf den News-Bereich. Der Key kann nicht allgemein zur Verfügung gestellt werden, da die Anzahl der Zugriffe beschränkt ist. Den Key hinterlegen Sie direkt in der Datei `NewsControl/ViewModel.cs` (Zeile 95).
- › **Weather:** Generieren Sie einen Key für den kostenfreien Zugriff und hinterlegen Sie diesen in der Datei `OpenWeatherMapProxyForecast.cs` (Zeile 13).

Das Einfügen der Keys für die Services von News und Wetter nehmen Sie direkt in Visual Studio vor. Kompilieren Sie das Projekt danach neu und erstellen Sie die App-Packages für die Installation auf dem Raspberry Pi.

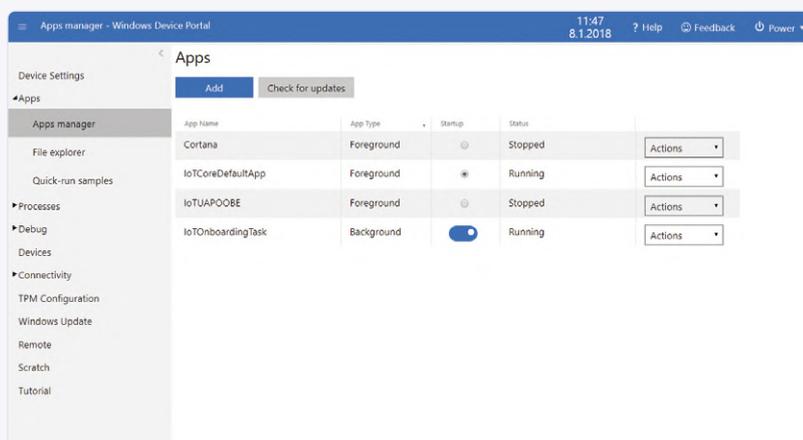


Bild B. Webinterface zur Konfiguration des Raspberry Pi.

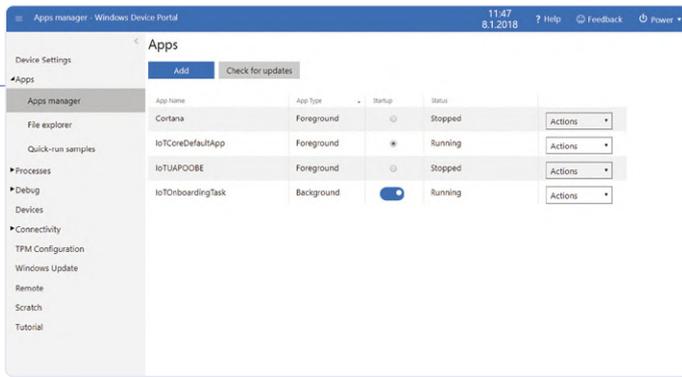


Bild 3. Optionen der OpenWeather-API.

entweder vom Anbieter des Service zur Verfügung gestellt oder ist als Bibliothek über die Community verfügbar. In Visual Studio wird eine SDK über den Paketmanager NuGet eingebunden.

► **Extern (REST):** Gibt es kein SDK, dann kann man jeden beliebigen Service über die allgemeine Web-Schnittstelle in der Regel als REST-Interface einbinden. Mit der üblichen Web-Methode (GET) sendet man eine Anfrage an den Dienst und erhält eine Antwort in einem strukturierten Datenformat (XML, JSON). Die Daten kann man nun parsen, interpretieren sowie nach seinen Wünschen grafisch aufbereiten und anzeigen. Cloud-Services, welche von Dritt-Applikationen genutzt werden können, bieten eine solche Schnittstelle und eine entsprechende Dokumentation. Bei der Nutzung von externen Services – gleichgültig ob über SDK oder REST – ist es meist Voraussetzung, dass man sich beim Dienst registriert. Zur Nutzung ist eine eindeutige Identifizierung notwendig. Dazu erhält man vom Dienst in der Regel einen Key, der bei jeder Abfrage zu übermitteln ist. Viele Services werden für eine private und eingeschränkte kommerzielle Nutzung kostenfrei angeboten, auch wenn dabei beispielsweise die Anzahl der Abrufe der Daten (Calls/Day) beschränkt ist oder man bezüglich der bereitgestellten Datenqualität Abstriche machen muss. Dennoch ist eine Vielzahl von Services

auch in der kostenfreien Variante sehr gut und umfassend nutzbar.

Services

Die App HomeInfoSystem besteht aus einer Kombination mehrerer Inhaltsoptionen (Services), welche man nach Belieben erweitern und kombinieren kann. In diesem Abschnitt beschreiben wir die Anbindung und Implementierung einiger Services. Daraus kann man Optionen zur Erweiterung ableiten:

► **Integration des Wetterdienstes:** Wir setzen hier auf die Verwendung des Wetterdienstes OpenWeather [3]. Dieser Dienst bietet unterschiedliche Services zum Abrufen der Wetterdaten. Neben kostenpflichtigen Angeboten kann man die Dienste *Current weather* und *5 days/3 hour forecast* kostenfrei nutzen (**Bild 3**). Nach der Registrierung generiert man einen API-Key, mit dem die Wetterdaten abgefragt werden können. Die Konvention eines solchen Aufrufs sieht zum Beispiel wie folgt aus:

```
api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key}
```

Dabei sind die Parameter `{city name}` und `{your api key}` entsprechend zu ersetzen. Die Antwort ist ein Datensatz im JSON-Format (**Listing 1**), den man auch gut „manuell“ lesen kann. Er beginnt mit den Koordinaten des Ortes, gefolgt von den Angaben zur Temperatur, Luftdruck, Bewölkung und so weiter. Der Datensatz wird auf der Webseite des Dienstes exakt beschrieben. Für die App wurde ein SDK entwickelt, welche eine einfache Nutzung des Service ermöglicht. In **Listing 2** sehen Sie den dazu gehörenden Abschnitt aus dem Quellcode.

► **Integration von Cloudspeicher:** Um Fotos darzustellen, greift man am besten auf einen der üblichen Cloudspeicher wie Dropbox oder OneDrive zu. Für einen Zugriff über externe Apps bieten

Listing 2. SDK (Ausschnitt) zur Nutzung des Service openweathermap.org.

```
// Hinweis: Der Datentyp RootObject enthält die Wetterdaten in einer Baumstruktur.

public async static Task<RootObject> GetWeather(string location)
{
    try
    {
        var httpClient = new HttpClient();
        string uri = basisUri + endPoint + "?q=" + location + "&appid=" + apiKey + "&units=metric";
        var response = await httpClient.GetAsync(uri);
        var result = await response.Content.ReadAsStringAsync();
        var serializer = new DataContractJsonSerializer(typeof(RootObject));
        var ms = new MemoryStream(Encoding.UTF8.GetBytes(result));
        var data = (RootObject)serializer.ReadObject(ms);
        return data;
    }
    catch
    {
        return null;
    }
}
```

diese Dienste ein eigenes API und darüber hinaus für viele Programmiersprachen auch ein SDK. Dropbox ermöglicht in der kostenfreien Variante 5 GB Speicherplatz für (beispielsweise) Bilder. Im Online-Portal können Sie direkt auf den Entwicklerbereich zugreifen. Richten Sie eine App bei Dropbox ein (**Bild 4**). Über das Portal können Sie die App konfigurieren, zum Beispiel auf welchen Ordner innerhalb der Dropbox-Struktur der Zugriff ermöglicht werden soll. Von außen gibt es mehrere Zugriffsmöglichkeiten auf die Daten (Fotos) in der Dropbox, über Nutzernamen und Passwort (Authentifizierung) oder ohne Nutzerinteraktion mit einem eindeutigen Schlüssel (Key, App Secret). Letztere Variante ist für uns optimal, da sich die App einfach mit einem solchen Key bei der Dropbox identifiziert. Wir nutzen das SDK für .NET Core, das wir direkt in unsere App integrieren können. Das Lesen der Fotos aus der Dropbox ist dann nur ein Kinderspiel: Den

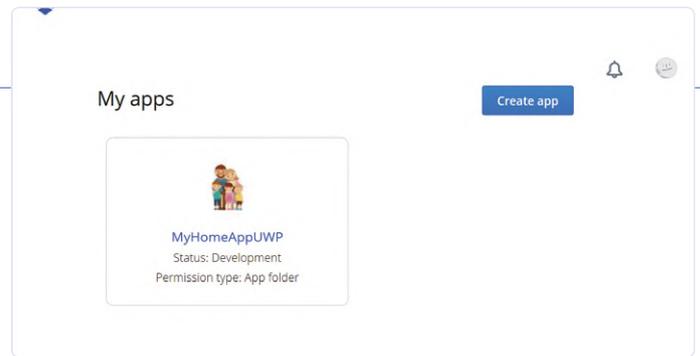


Bild 4. Im Entwicklerportal von Dropbox muss man eine App einrichten.

entsprechenden Codeabschnitt sehen Sie in **Listing 3**. Anwender können zum Beispiel Fotos mit dem Smartphone aufnehmen und direkt aus dem Mobilgerät in der Dropbox speichern.

Listing 3. SDK (Ausschnitt) zum Zugriff auf die Daten in der Dropbox.

```
public static async Task<ObservableCollection<string>> GetFilesFromDropBoxAsync()
{
    ObservableCollection<string> itemList = new ObservableCollection<string>();
    // Pfad für Speicherung der Datei
    StorageFolder storageFolder = ApplicationData.Current.LocalFolder;
    try
    {
        using (var dbx = new DropboxClient(ProgrammSettings.DropBoxAppToken))
        {
            // Liste aller Dateien im App-Ordner
            var list = await dbx.Files.ListFolderAsync(string.Empty);
            // Schleife über alle Dateien
            for (int i = 0; i < list.Entries.Count; i++)
            {
                string localFileName = storageFolder.Path + "/" + list.Entries[i].Name;
                if (File.Exists(localFileName) == false)
                {
                    // Datei herunterladen
                    var file = await dbx.Files.DownloadAsync(list.Entries[i].PathLower);
                    // Datenstream extrahieren
                    Stream stream = await file.GetContentAsStreamAsync();

                    // Datenstream in lokale Datei schreiben
                    using (var fileStream = File.Create(localFileName))
                    {
                        (await file.GetContentAsStreamAsync()).CopyTo(fileStream);
                    }
                }
                itemList.Add(localFileName);
            }
        }
    }
    catch
    {
    }
    return itemList;
}
```

- **Statische Inhalte:** Statische Inhalte werden mit der Programmlogik direkt in der App implementiert. Dazu gehört beispielsweise die Anzeige der Zeit mit einer besonderen visuellen Darstellung.

Diese Aufzählung zeigt beispielhaft Implementierungen von Inhalten für die App HomeInfoSystem. Für Anwendungen in Unternehmen (EnterpriseInfoSystem) muss man für bestimmte spezifische Services ebensolche Anbindungen schaffen.

Erweiterungen

Die zuvor beschriebenen Services sind lediglich Beispiele und sollen Sie zum Start eines eigenen HIS-Projekts animieren. Sie werden sicher schnell eigene Services finden, die Sie implementieren möchten. Aus meiner Sicht kann man das System sinnvoll erweitern durch:

- Das HomeInfoSystem könnte auch die Werte von verschiedenen Sensoren anzeigen. Die Werte der Sensoren werden an ein IoT-Backend übertragen und von der App in bestimmten Intervallen abgegriffen. Elektroniker können umfassende Hardwareerweiterungen bauen und diese in das System flexibel integrieren.
- **Inhalte mit Zeitbezug einblenden:** Einige Inhalte können mit einem Zeitbezug versehen werden. So ist es sinnvoll, Wochentags zwischen 7 Uhr und 8 Uhr die aktuellen Informationen zu lokalen Nahverkehrsverbindungen einzublenden. Am Wochenende dagegen ist man vielleicht eher an Informationen zu aktuellen Filmen im örtlichen Kino interessiert, die man über eine Kinoprogramm-API ermittelt.
- **Mobile Services:** Man könnte die Systemumgebung um eine mobile App erweitern. Nutzer können Daten an einen Server (Backend) senden und die App ruft diese Daten ab. Dabei sind vielfältige Anwendungen denkbar. Eine weitere spannende Anwendung ist die Nutzung von Geoservices. Mobile Anwender können zum Beispiel (nach Zustimmung und Freigabe) ihre aktuelle Geoposition an ein Backend übermitteln. Die App ruft diese Position regelmäßig ab und zeigt diese auf einer Karte an. Auf diese Weise weiß man, wo sich alle Mitglieder einer Familie gerade befinden.
- **Online Services:** Weitere interessante Services, die man über eine öffentliche API einbinden kann, sind: Kalenderintegration verschiedener Dienste wie Microsoft- oder Google-Kalender, Fahrplan der Deutschen Bahn oder des Verkehrsverbundes (nach Angabe des Ortes für die Abfrage der Verbindungsdaten).
- **Alternative Interface-Struktur:** Statt eines manuellen oder zeitgesteuerten Wechsels zwischen Inhalten könnte man auch eine Gesamtansicht erstellen. Das dürfte insbesondere für größere Bildschirme interessant sein. Beispielsweise könnte man auf der einen Seite das aktuelle Wetter anzeigen und auf der anderen Seite im Wechsel die Nachrichten und aktuelle Fotos.

Sie sehen, es gibt genügend Ideen, um das HomeInfoSystem zu erweitern. Ein besonderer Reiz liegt darin, externe Sensoren (IoT) anzubinden und so zum Beispiel kurz einmal einen Blick auf das Ferienhaus (Kamera) zu werfen. Im unternehmerischen Kontext dürfte es ebenso genügend Anwendungsmöglichkeiten geben.

Fazit und Ausblick

Ein solches Infosystem im Heimbereich wie auch in einem Unternehmen ist eine faszinierende Sache. Zu Hause kann es aktuelle Informationen liefern (Wetter, News, Termine) und im Unternehmen wichtige aktuelle Daten präsentieren (Stückzahlen, Termin Teammeeting). Das Interessante ist jedoch, dass man es mit überschaubarem Aufwand an die eigenen Bedürfnisse anpassen kann. Nach der Auswahl der Hardwarekonfiguration, insbesondere des Bildschirms, steht mit der hier beschriebenen Software bereits ein Grundstock an Funktionen zur Verfügung, welche man dann flexibel erweitern und anpassen kann. Der Quellcode ist Open Source. Sie können den Code [1] in Visual Studio 2019 öffnen und nach Ihren eigenen Bedürfnissen weiterentwickeln und modifizieren. ❏

191247-01



IM ELEKTOR-STORE

➤ Raspberry Pi 3 B+

www.elektor.de/raspberry-pi-3-model-b-plus



➤ Elektor Raspberry Pi Elektronik-Kit

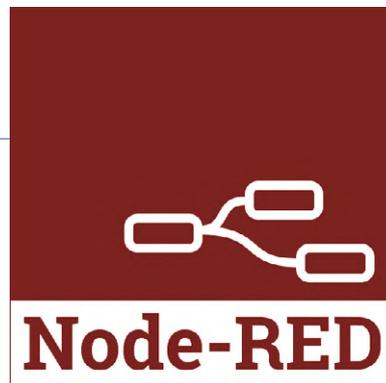
www.elektor.de/elektor-raspberry-pi-elektronik-kit

WEBLINKS

- [1] Projektsoftware: www.elektormagazine.de/191247-01
- [2] VisualStudio: <https://visualstudio.microsoft.com/downloads/>
- [3] OpenWeather: <https://openweathermap.org/>
- [4] Windows 10 IoT-Core: <https://docs.microsoft.com/en-us/windows/iot-core/downloads>
- [5] News-API: <https://newsapi.org/>

Einstieg in Node-RED

Ein visuelles blockbasiertes Open-Source-Programmierwerkzeug



Von **Dogan Ibrahim**

Node-RED ist in letzter Zeit sehr populär geworden, da es die Entwicklung von IoT-Projekten erheblich vereinfacht. In diesem Artikel sehen wir, was es mit Node-RED auf sich hat und am Beispiel eines sehr einfachen Projekts wie der Steuerung einer LED, wie es eingesetzt werden kann.

In den letzten Jahren haben die Entwicklung und der Einsatz von blockbasierten visuellen Programmierwerkzeugen einen starken Aufschwung erfahren. Die Idee dahinter ist, dass dem Programmierer ein Satz von visuellen Blöcken zur Verfügung gestellt wird, die er lediglich auf logische Weise miteinander verbinden muss, um das gewünschte Anwendungsprogramm zu erstellen. Node-RED ist ein solches visuelles blockbasiertes Programmierwerkzeug, das kostenlos und open-source ist, aber dennoch die Ausführung sehr komplexer Aufgaben erlaubt, einschließlich Web-Zugang, Twitter, E-Mail, HTTP-Abfragen, Bluetooth, MQTT sowie Steuerung der GPIO-Ports gängiger Computer wie Raspberry Pi, Arduino und ESP32. Das Schöne an Node-RED ist, dass der Programmierer nicht lernen muss, komplexe Programme zu schreiben. Zum Beispiel kann er ein Programm zum Verschicken einer E-Mail erstellen, indem er ein paar Knoten zusammenfügt und nur ein paar Zeilen einfachsten Code schreibt.

Node-RED, die Inbetriebnahme

Node-RED ist bereits im Raspbian-Betriebssystem des Raspberry Pi enthalten und kann entweder vom Desktop (**Bild 1**) oder durch

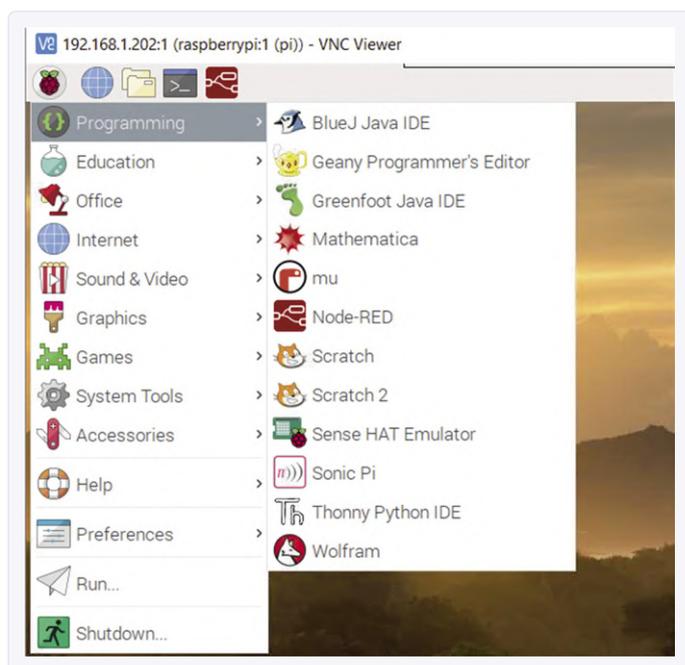


Bild 1. Es beginnt ganz einfach: Node-RED auf dem Raspberry Pi.

Eingabe des Befehls

```
pi@raspberrypi:~ $ node-red-start
```

von der Befehlszeile (**Bild 2**) aus gestartet werden.

Node-RED wird als Service gestartet. Eine kurze Liste gültiger Befehle zum Starten und Stoppen dieses Dienstes wird nach dem Start auf dem Bildschirm angezeigt, wie es unten in Bild 2 schon dargestellt ist. Die lokale IP-Adresse 192.168.1.202 ist die des Raspberry Pi des Autors. Sie können die IP-Adresse Ihres Raspberry Pi leicht finden, indem Sie den Befehl `ifconfig` am Prompt eingeben und im Ergebnis auf die Zeile schauen, die mit `wlan0` beginnt.

Wenn Sie „Ihre“ IP-Adresse wissen, können Sie sie im Browser Ihres PCs eingeben, um das grafische Node-RED-Benutzerinterface auf dem Monitor zu starten:

```
https://<Ihre IP-Adresse>:1880
```

Beim Autor beispielsweise wäre die Eingabe im Browser:

```
https://192.168.1.202:1880
```

Bild 3 zeigt den dreiteiligen Startbildschirm von Node-RED. Auf der linken Seite befindet sich eine Liste der Knoten, die bei der Installation des Programms standardmäßig verfügbar sind, sich aber bei Bedarf mit zusätzlichen Nodes aus verschiedenen Quellen im Internet erweitern lässt. Der mittlere Teil ist der Arbeitsbereich, in dem die erforderlichen Knoten per Drag & Drop gezogen und zu einem Ablaufprogramm verbunden werden. Auf der rechten Seite befindet sich der Informations- und Debug-Bereich, der bei der Programmentwicklung sehr nützlich werden kann.

Schauen wir uns den Umgang mit Node-RED anhand eines sehr einfachen Beispiels an, einer LED, die über einen 220-Ω-Strombegrenzungswiderstand an GPIO 2 des Raspberry Pi angeschlossen ist (**Bild 4**). Die LED soll von einem Node-RED-Flussprogramm gesteuert werden. Dieses lächerlich einfache Flussprogramm (**Bild 5**) wird so aufgebaut:



Bild 2. Starten von Node-RED von der Befehlszeile aus.

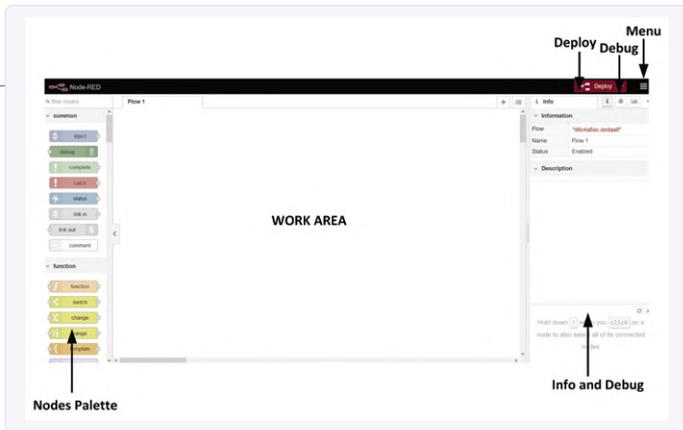


Bild 3. Startmenü von Node-RED.

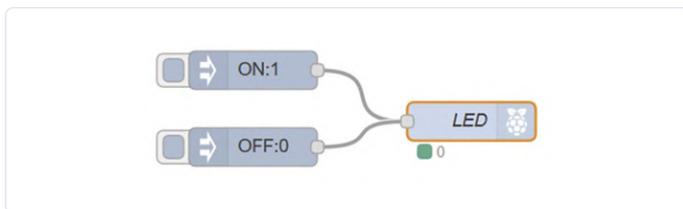


Bild 5. Das Flussdiagramm des Experiments.

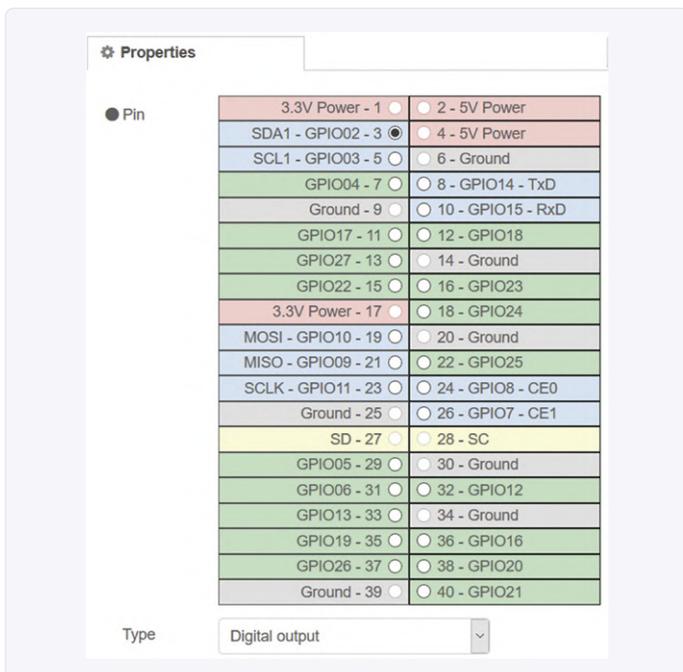


Bild 6. Konfigurieren des Knotens *rpi gpio out*.

- Der *Inject*-Knoten wird angeklickt, in den Arbeitsbereich gezogen und abgelegt. Mit Injektionsknoten werden Nachrichten (wie Strings oder Zahlen) in ein Flussprogramm eingebracht. In diesem Beispiel werden wir 1 oder 0 „injizieren“, um die LED ein- und auszuschalten.
- Doppelklicken Sie auf den Injektionsknoten, um ihn zu konfigurieren. Dabei wird die *Payload* auf *number* gesetzt und eine „1“ eingegeben. Durch Klicken auf das quadratische Feld am linken Rand dieses Knotens wird „1“ ausgegeben, um die LED einzuschalten.
- Stellen Sie *Topic* des Injektionsmodus auf ON, so dass Sie die Funktion dieses Knotens sehen können. Klicken Sie dann auf *Done*, um das Konfigurationsfenster zu schließen.
- Klicken, ziehen und legen Sie einen weiteren *Inject*-Knoten in den Arbeitsbereich, setzen Sie *Payload* auf „0“ und das *Topic* auf OFF. Wenn Sie auf diesen Knoten klicken, wird eine „0“ ausgegeben, um die LED auszuschalten.
- Legen Sie nun einen Knoten *rpi gpio out* in den Arbeitsbereich. Dieser Knoten sendet digitale oder PWM-Daten an einen GPIO-Port des Raspberry Pi. Schließen Sie den Knoten wie in Bild 5 an.
- Durch einen Doppelklick auf diesen Knoten erscheint ein Fenster (Bild 6), in dem Sie den GPIO-Anschluss des Raspberry Pi konfigurieren können. Setzen Sie *Pin* auf GPIO2, *Type* auf *Digital output*, klicken Sie auf *Initialise pin state* und setzen Sie den Anfangszustand der LED auf „0“. Setzen Sie den Namen des Knotens auf *LED*. Abschließend klicken Sie auf *Done*, um den Konfigurationsbildschirm zu schließen.

Damit ist das Flussprogramm abgeschlossen. Klicken Sie auf *Deploy*, um das Programm zu kompilieren und einen ausführbaren Code zu generieren. Vergewissern Sie sich, ob in diesem Stadium des Entwurfs keine Fehlermeldungen mehr angezeigt werden, dann dürfen Sie auf die quadratische Schaltfläche auf der linken Seite des *ON*-Injektionsknotens klicken, um die LED einzuschalten. Wenn Sie dann auf die Schaltfläche des anderen Injektionsknotens klicken, wird die LED wieder dunkel.

Weitere Aspekte

An diesem sehr einfachen Beispiel haben wir gesehen, was Node-RED ist und wie es verwendet wird. Eine der Stärken von Node-RED ist, dass es von großen Communities unterstützt wird und dass es viele von Dritten entwickelte Nodes gibt, die kostenlos zur Verfügung stehen und in komplexen Projekten eingesetzt werden können. Zum Beispiel gibt es den Knoten *openweathermap*, der die lokalen Wetterbedingungen wie Temperatur, Feuchtigkeit, Luftdruck, Windgeschwindigkeit, Niederschlagsmenge und so weiter überall auf der Erde angibt. Ein Projekt mit einer Wettervorhersage lässt sich mit diesem und einigen zusätzlichen Knoten in weniger als einer Stunde erstellen. Würde man ein solches Projekt auf herkömmliche Weise mit „eigenen“ externen Sensoren realisieren, wäre dies viel komplizierter, zeitaufwändiger

WEBLINKS

- [1] **Offizielle Website von Node-RED:** <https://nodered.org>
- [2] **Erste Schritte:** <https://projects.raspberrypi.org/en/projects/getting-started-with-node-red>
- [3] **Ein erstes Flussprogramm:** <https://nodered.org/docs/tutorials/first-flow>
- [4] **Schreibfunktionen:** <https://nodered.org/docs/user-guide/writing-functions>
- [5] **Programmierhandbuch:** <http://noderedguide.com/examples/>

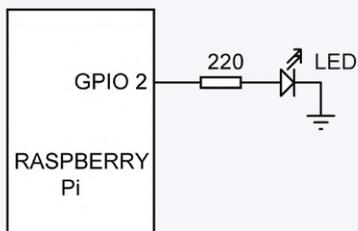


Bild 4. Der Schaltplan des Experiments.



IM ELEKTOR-STORE

Ein neues Buch des Autors mit dem Titel *With Node-RED: Design IoT Projects with Raspberry Pi, Arduino and ESP32* enthält eine große Zahl von Node-RED-Projekten mit den derzeit populärsten Mikrocontroller-Entwicklungsboards.

- > **Das Buch kann im Elektor Store bestellt werden unter:**
www.elektor.de/programming-with-node-red
- > **Ebenfalls erhältlich ist eine E-Book-Version:**
www.elektor.de/programming-with-node-red-e-book

und in der Regel auch erheblich kostspieliger. Node-RED unterstützt UDP- und TCP-basierte Kommunikation mit einem Knoten über eine WLAN-Verbindung. Diese Funktion ermöglicht es Benutzern, drahtlos Daten von entfernten Stellen über eine WLAN-Verbindung von vernetzten Geräten wie Mobiltelefonen, Tablets oder PCs zu senden und zu empfangen. Einige andere interessante Knoten, die kostenlos aus dem Internet installiert werden können, sind:

- > Dashboard
- > Bluetooth
- > Amazons Alexa
- > I²C- und paralleles LCD
- > Weltkarte
- > A/D-Wandler
- > Ultraschall-Sensor

Node-RED ist nicht nur für die den Raspberry Pi gedacht, sondern

kann (wenn auch eingeschränkt) mit den Prozessorboards Arduino und ESP32 oder Kombinationen von Raspberry Pi und Arduino oder Raspberry Pi und ESP32 verwendet werden.

Einer der sehr wichtigen Node-RED-Knoten ist der *function*-Knoten, der so konfiguriert werden kann, dass er mehrere Ausgänge hat. Benutzer können bei der Konfiguration dieses Knotens passende Programmcodes schreiben. So lassen sich mit dem *function*-Knoten aus den lokalen Wetterdaten, die der *openweathermap*-Knoten liefert, beispielsweise die lokalen Temperatur- und Feuchtigkeitsdaten extrahieren. Dies kann man leicht durch ein paar Befehle bei der Konfiguration des *function*-Knotens erreichen. Interessierte Leser bekommen weitere Infos unter den Weblinks [1]..[5] oder im Node-RED-Buch des Autors. ◀

200164-03

Anzeige



Kunststoffgehäuse Standard und Miniatur

Mehr als 5000 verschiedene Gehäusedesigns.
hammfg.com/electronics/small-case

+ 44 1256 812812
sales@hammondmfg.eu



ONE STEP FOR US AND X PLUS FOR YOU



Setup für Bode-Plot Messung

Bode Plot & opt.
Power Analysis

Opt. Built-in
50 MHz AWG

10 - bit
Resolution Mode

200 Mpts/ch
Memory Depth

100 / 200 / 350
MHz Bandwidth

2 GSa/s
Sampling Rate

SDS2000X PLUS Oscilloscopes

Die **e**lektor Sommer- Präsente



Unsere Freunde von NextIND ziehen alle Register!

- > 2 x TACTIGON-ONE-Board
- > 1 x Kit bestehend aus einem Raspberry- und drei TACTIGON-ONE-Boards
- > 1 x TACTIGON-SKIN-Entwickler-Kit (ohne Roboter)



Angeboten von Gudeco:

- > 2 x Lötstation Weller WT 1010



iFixit bietet nicht ein, nicht zwei, ... nicht vier, sondern:

- > 5 x Pro-Tech-Toolkit



Simac / Joy-IT lässt die Sonne scheinen:

- > 1 x DC-Stromversorgungs-Set JOY-iT JT-RD6006
- > 1 x 3-in-1-Handheld-Oszilloskop, Signalgenerator und Multimeter DMSO2D72



Der Franzis Verlag spendierte drei (deutsche) Kits:

- > 1 x MAKER KIT für ESP32
- > 1 x MAKER KIT für SMART HOME
- > 1 x MAKER KIT für ARDUINO®



Von Peak Electronic Design kam:

- > 1 x ZEN50 – der Zenerdioden-Analysator Atlas ZEN
- > 1 x ESR70 – das Messgerät für den Äquivalenten Serienwiderstand Atlas ESR PLUS



LabNation bietet Ihnen an:

- > 1 x SmartScope - Das Oszilloskop der neuen Generation



The MagPi – die offizielle Raspberry-Pi-Zeitschrift – lässt sich nicht lumpen:

- > 1 x Raspberry Pi 4 – 8 GB
- > 1 x Hochwertiges Kameramodul
- > 1 x Kameraobjektiv 6 mm CS-Mount

Wie kommt man an diese Give-aways?

Möchten Sie eines dieser Produkte erhalten? Besuchen Sie unsere Website und teilen Sie uns einfach Ihre Vorliebe mit. Wir werden die glücklichen Gewinner in der letzten Augustwoche ermitteln. Weitere Informationen über die Produkte und diese kleine Give-away-Aktion finden Sie hier:

www.elektor.de/give-away



Diese Präsente werden stolz präsentiert von



Einfacher Funktionsgenerator



Mit umgekehrter Signalerzeugung

Von **Michael A. Shustov** (Russland) und **Andrey M. Shustov** (D)

INFOS ZUM PROJEKT

Stichworte

Funktionsgenerator, Sinus, Dreieck, Rechteck, Labor, Analog, Opamp

Niveau

Einsteiger – Fortgeschrittene – Experte

Zeit

Etwa zwei Stunden

Werkzeuge

LötKolben

Kosten

ca. 35 €

EIGENSCHAFTEN

- › Sinus-, Dreieck- und Rechtecksignale
- › Signalgenerierung in „umgekehrter“ Reihenfolge
- › Sinuswellenbereiche 50...500 Hz, 0,5...4 kHz
- › Dreieck- und Rechteckwellenbereiche 100...1.000 Hz, 1...8 kHz
- › Frequenzbereiche leicht anpassbar
- › Low-Budget-Projekt
- › Nur bedrahtete Bauteile
- › 100% mikrocontrollerfrei

Ältere Signalgeneratoren erzeugen Sinus-, Dreieck- und Rechteckwellen normalerweise durch RC-Generatoren, gefolgt von geeigneten Filtern. Alternativ und sicherlich moderner können solche Signale auch durch einen Mikroprozessor synthetisiert werden. Der hier beschriebene Funktionsgenerator funktioniert völlig anders!

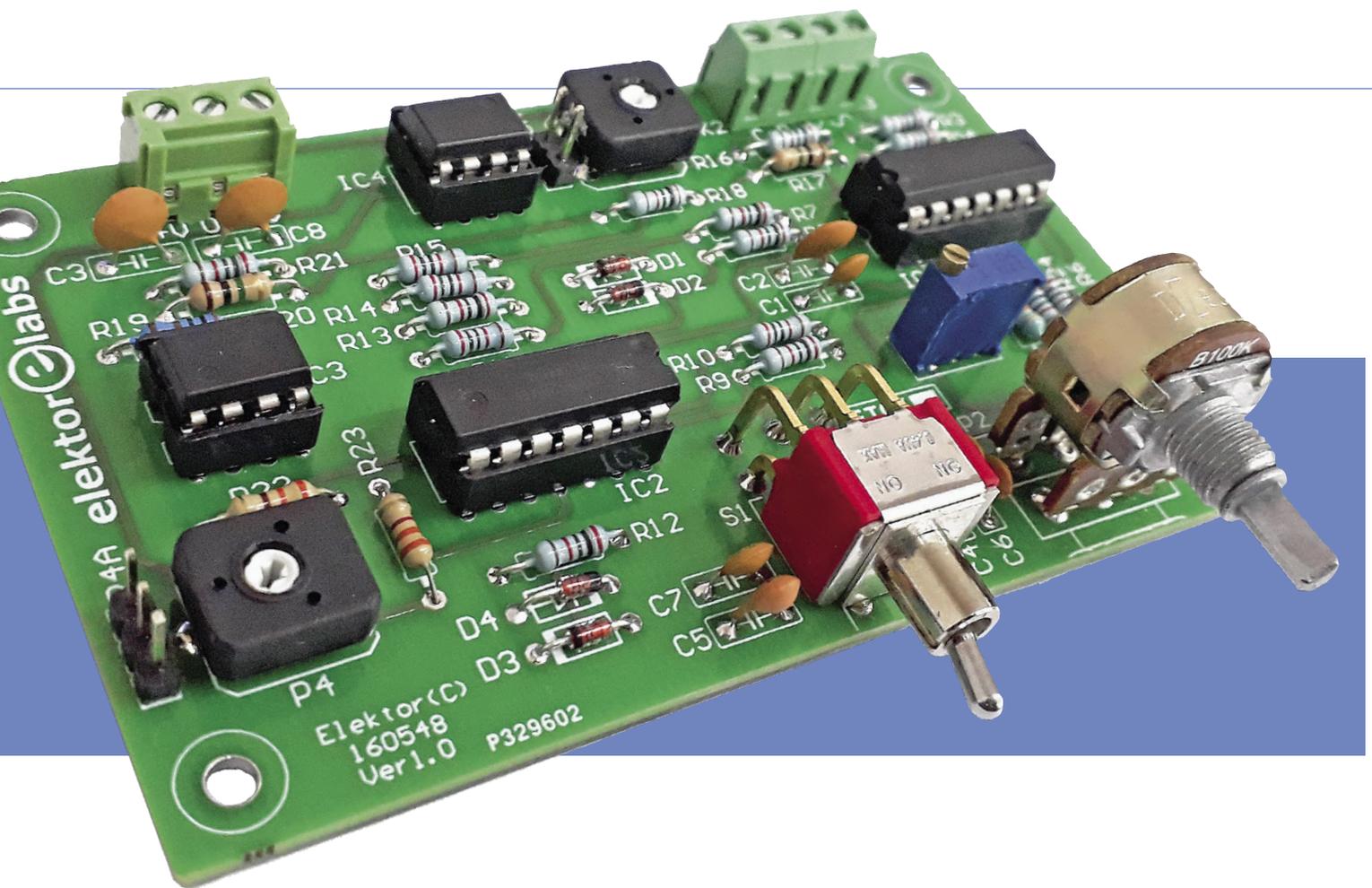
Ein Funktionsgenerator ist ein Gerät, das üblicherweise mehr als eine Wellenform variabler Frequenz und Amplitude erzeugt. Damit kann man Signale generieren, die im Labor zum Testen, Optimieren, Fehlersuchen, Reparieren und Abstimmen anderer elektronischer Geräte benötigt werden.

In den meisten „klassischen“ Funktionsgeneratoren ist ein Rechteckimpulsgenerator das zentrale Element. Als nächstes kommt dann ein Rechteck-Dreieck-Wandler, der normalerweise auf der Ladung/Entladung eines Kondensators basiert. Zu guter Letzt wird dann diese dreieckige Kurvenform elektronisch halbwegs akzeptabel in eine Sinuswelle umgewandelt, was eine gute Unterdrückung der ersten Harmonischen einschließt. Der Hauptnachteil dieses dreistufigen Prozes-

ses und der damit verbundenen Schaltung ist die inhärente Nichtlinearität der Lade/Entlade-Prozesse von Kondensatoren, die sich besonders bemerkbar macht, wenn die Frequenz des Generators verändert wird. Besonders relevant ist auch die deutliche Verzerrung des Sinussignals, infolge suboptimaler Filterung der höheren Harmonischen eines komplexen Signals.

Umgekehrt

Der unten beschriebene Funktionsgenerator ist insofern unüblich, als hier die Umwandlung von Signalformen in (gegenüber dem Üblichen) umgekehrter Reihenfolge erfolgt. Zuerst wird ein sinusförmiges Signal erzeugt, das dann in ein Dreieckssignal umgewandelt wird. Zum Schluss erst wird aus dem dreiecki-



gen Signal eine Rechteckwelle geformt. Der Generator verfügt über zwei umschaltbare Frequenzbänder. Das Sinussignal wird von einem Oszillator auf Opamp-Basis erzeugt. Das Dreieckssignal wird durch Gleichrichtung und Invertierung des Sinuswellensignals generiert. Schließlich wird daraus ein Rechtecksignal durch sehr hohe Verstärkung gewonnen.

Schaltung

Von der Theorie zur Praxis: **Bild 1** zeigt die Schaltung des einfachen Funktionsgenerators. Sie arbeitet mit einer symmetrischen $\pm 5\text{-V}$ -Versorgung. Die Opamps IC1.A, IC1.B und IC1.C bilden den Sinusoszillator. Er gibt zwei Sinuswellen mit einer Phasendifferenz von 90° aus. Die Lage der beiden Frequenzbänder hängt von den Werten von C4 und C6 (33 nF) bzw. von C5 und C7 (3,3 nF) ab. Die Rückkopplung des Oszillators erfolgt über R24. Trimpoti P1 dient zur Einstellung des Rückkopplungspegels auf den Wert, bei dem sich eine nicht begrenzte Sinuswelle einstellt. Innerhalb des gewählten Bereichs wird die Generatorfrequenz mit dem Stereo-Poti P2 eingestellt. S1 schaltet zwischen den zwei Frequenzbereichen um. C4 plus C6 sind für den niedrigen und C5 plus C7 für den hohen Bereich zuständig.

SCHALTUNG & MATHEMATIK

Die drei verschiedenen hier vorkommenden Wellenformen können durch die folgenden Formeln beschrieben werden. Die tiefgestellten Indizes des Ausdrucks $U_{A-G}(t)$ entsprechen den Punkten in der Schaltung, die mit eingekreisten Buchstaben gekennzeichnet sind.

$$U_A(t) = U_O \sin(\omega t)$$

$$U_B(t) = U_O \sin(\omega t + 90^\circ)$$

$$U_C(t) = U_O \sin(2\omega t)$$

$$U_D(t) = U_O \text{abs}[\sin(\omega t)]$$

$$U_E(t) = -U_O \text{abs}[\sin(\omega t + 90^\circ)]$$

$$U_F(t) = U_O \{ \text{abs}[\sin(\omega t)] - \text{abs}[\sin(\omega t + 90^\circ)] \}$$

$$U_G(t) = \begin{cases} -U_O & \text{if } U_F(t) > 0 \\ U_O & \text{if } U_F(t) < 0 \end{cases}$$

Die Signale A OUT und B OUT werden den beiden identischen Gleichrichterschaltungen aus IC2.A/B und IC2.D/C zugeführt, die als

Vollweggleichrichter fungieren. Auch die beiden gleichgerichteten Signale sind um 90° phasenverschoben – haben aber die doppelte Frequenz

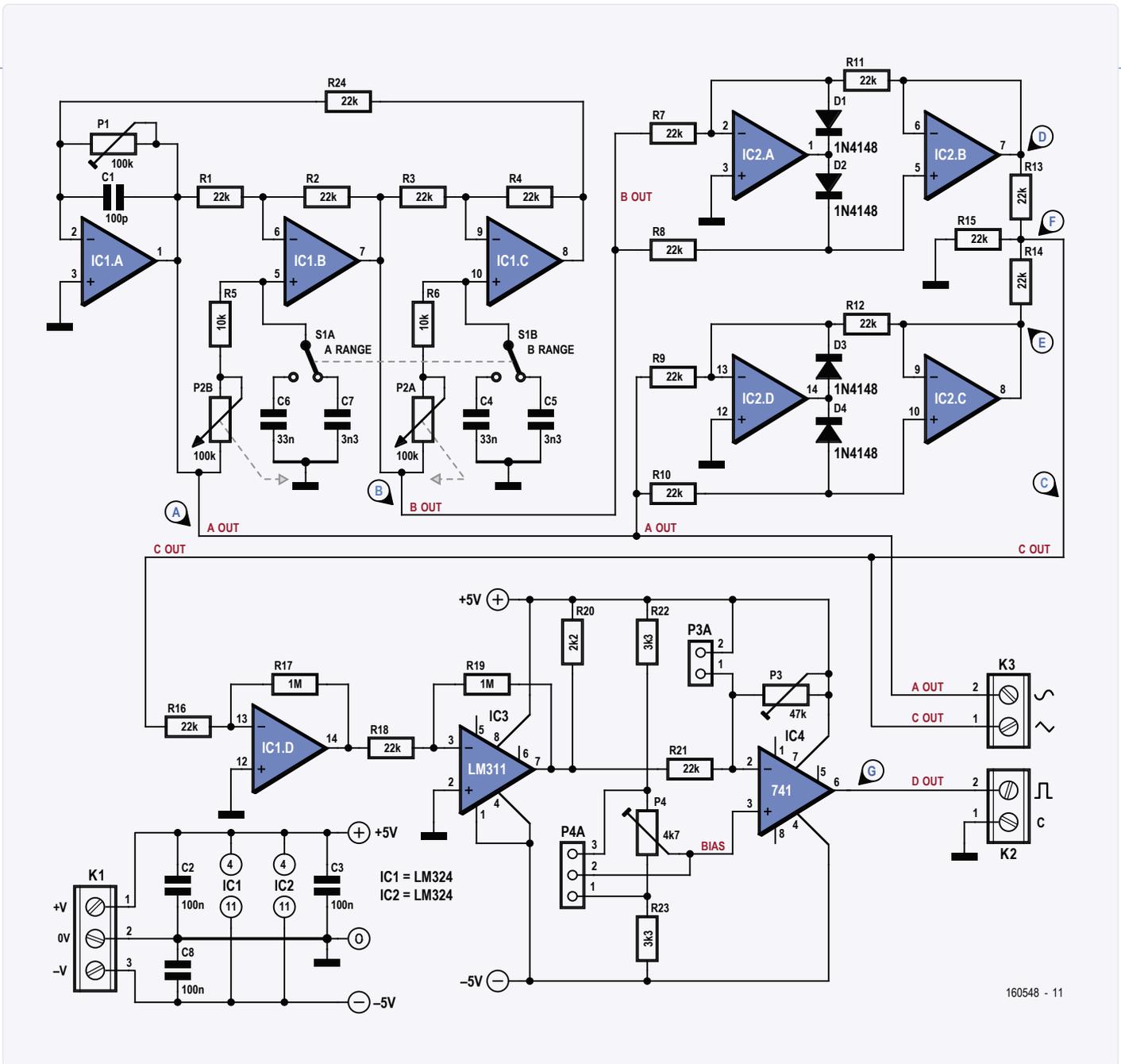


Bild 1. Schaltplan des einfachen Funktionsgenerators mit Bezeichnungen der wichtigsten internen Signale (siehe Kasten Schaltung & Mathematik). Ungewöhnlich ist die Reihenfolge der Signalgenerierung: Sinus → Dreieck → Rechteck.

des steuernden Sinussignals. Die gleichgerichteten Signale werden zum Signal C OUT kombiniert. Diese (mathematische) Addition ergibt ein Dreieckssignal – wenn auch mit geringerer Amplitude. Das Dreieckssignal gelangt dann an IC1.D (der restliche Opamp des LM324). Zusammen mit IC3 wird durch massive Verstärkung daraus ein Rechtecksignal. Mit Trimpoti P4 und IC4 wird die Offset-Spannung des Rechtecksignals beeinflusst und über Trimpoti P3 die Amplitude des Signals D OUT eingestellt. Alle drei Ausgangssignale stehen an den beiden zweipoligen Schraubklemmen K2 und K3 zur Verfügung.

Wenn S1 auf den niedrigen Bereich eingestellt ist (C4/C6), beträgt der mit P2 einstellbare

Frequenzbereich 50...500 Hz für das Sinussignal bzw. 100...1.000 Hz für das Dreieck- und Rechtecksignal (Verdoppelung der Sinusfrequenz).

Wenn S1 auf C5/C7 eingestellt ist, verzehnfacht sich die Frequenz auf einen Bereich von 0,5...5 kHz für das Sinussignal und 1...10 kHz für Dreiecks- und Rechtecksignale – leider nur theoretisch. In der Praxis ergeben sich etwa 0,5...4 kHz bzw. 1...8 kHz.

Durch Modifizieren der frequenzbestimmenden Kondensatoren oder Hinzufügen zusätzlicher Bereiche mit mehrstufigem S1 können Frequenzen bis in den Sub-Hz-Bereich hinab generiert werden. Auch höhere Frequenzen als angegeben sind möglich,

doch die begrenzte Geschwindigkeit der Opamps begrenzt die Brauchbarkeit bei deutlich höheren Frequenzen.

Aufbau

Elektor hat eine Platine für den einfachen Funktionsgenerator entwickelt. Bild 2 zeigt, wie der fertig aufgebaute Prototyp aussieht. Die Bestückung sollte auch für Anfänger relativ leicht machbar sein, da nur bedrahtete Bauteile verwendet werden und kein Mikrocontroller vorhanden ist.

Die mit P3A und P4A gekennzeichneten Stiftheisten ermöglichen den Einsatz richtiger externer Potis für Offset und Pegel. In diesem Fall werden P3 und P4 nicht bestückt.



STÜCKLISTE

Widerstände:

R1..R4,R7..R16,R18,R21,R24 = 22 k

R5,R6 = 10 k

R17,R19 = 1 M

R20 = 2k2

R22,R23 = 3k3

P1 = 100 k, Mehrgang-Trimpoti, stehend

P2 = 100 k, Stereo-Poti, linear

P3 = 47 k, Trimpoti, klein, liegend

P4 = 4k7, Trimpoti, klein, liegend

Kondensatoren:

C1 = 100 p / 50 V, C0G, 5 %

C2,C3,C8 = 100 n / 50 V, keramisch, MCFY

C4,C6 = 33 n / 100 V, SkyCap SR, 10 %

C5, C7 = 3n3 / 50 V, C0G, 5 %

Halbleiter:

D1-D4 = 1N4148

IC1,IC2 = LM324

IC3 = LM311

IC4 = LM741 (μ A741)

Außerdem:

K1 = 3-pol. Schraubklemme,

Platinenmontage, RM 3,5 mm

K2,K3 = 2-pol. Schraubklemme,

Platinenmontage, RM 3,5 mm

P3A = 2-pol. Stiflleiste (1x2), gerade, RM 0,1"

P4A = 3-pol. Stiflleiste (1x3), gerade, RM 0,1"

S1 = Umschalter, doppelpolig

2x 8-pol. IC-Fassung, DIP

2x 14-pol. IC-Fassung, DIP

Test

Wenn die Platine erfolgreich bestückt wurde, wird sie nach folgendem Procedere getestet:

- › Verbinden der ± 5 -V-Versorgung mit K1.
- › Auswahl des Frequenzbereichs (niedrig oder hoch) mit S1.
- › Anschluss eines Oszilloskops an A OUT (K3, Sinus) und C (K2, GND).
- › Mit P1 ein möglichst sauberes Sinussignal einstellen.
- › Verbinden des Oszilloskops mit C OUT (K3). Ist ein Dreiecksignal vorhanden?
- › Verbinden des Oszilloskops mit D OUT (K2). Ist ein Rechtecksignal vorhanden?
- › Mit P2 prüfen, ob sich die Frequenz korrekt verstellen lässt.
- › Mit P3 die Amplitude des Signals einstellen.
- › Mit P4 die Rechteck-Offsetspannung einstellen.

Das war's auch schon. Jetzt haben Sie einen fertigen Funktionsgenerator für wenig Geld, mit dem sich viele andere Geräte mit sinus-, dreieck- oder rechteckförmigen Signalen versorgen lassen, um deren Funktion zu testen. Viel Spaß damit! ◀

(160548)

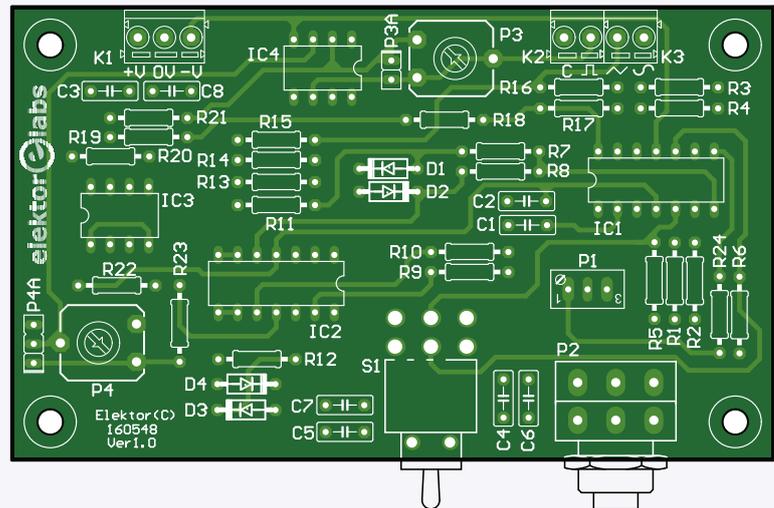
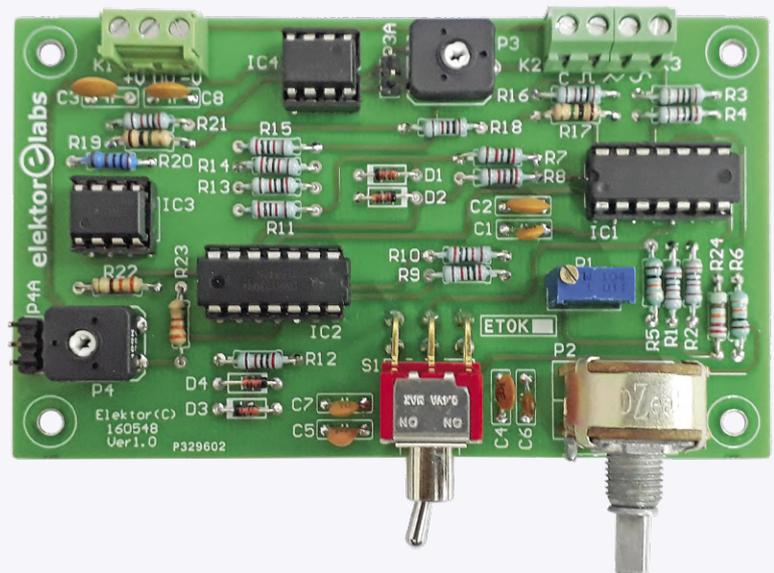


Bild 2. Prototyp des einfachen Funktionsgenerators. Es sind keine SMDs zu sehen.



IM ELEKTOR-STORE

› Einfacher Funktionsgenerator, unbestückte Platine

www.elektor.de/function-generator

› Buch: Electronics for All

www.elektor.de/electronic-circuits-for-all

› Buch: Analog Circuit Design, Volume 1

www.elektor.de/analog-circuit-design-1

› Buch: Analog Circuit Design, Volume 2

www.elektor.de/analog-circuit-design-2

› Buch: Analog Circuit Design, Volume 3

www.elektor.de/analog-circuit-design-3



baut ein LoRa- Alarmsystem



Link zu
YouTube

von **GreatScott!**, mit einem Intro & Outro von **Luc Lemmens** (Elektor-Labor)

In diesem Artikel stellen wir ein Projekt vor, das von Youtuber GreatScott! entworfen wurde, um mit LoRa eine Meldung des Einbruchdetektors in seiner 600 m entfernten Garage in seine Wohnung zu übertragen. Er verwendete zwei STM32-Nucleo-Boards und zwei RFM95-LoRa-Module auf Elektor-Breakout-Boards (die von Elektor gestiftet wurden), um die Hardware zu bauen und zu testen.

Laut Wikipedia ist „Great Scott“ ein in der zweiten Hälfte des 19. und zu Beginn des 20. Jahrhunderts in den USA populärer Ausruf der Überraschung, des Erstaunens oder der Bestürzung. Wer „Zurück in die Zukunft“ im Original gesehen hat, wird sich vielleicht daran erinnern, dass Emmett „Doc“ Brown den Ausdruck oft verwendete, aber heutzutage kennen Sie ihn vielleicht noch besser vom YouTube-Kanal GreatScott!, der seit dem Jahr 2013 zahlreiche Elektronikprojekte und Tutorials vorgestellt hat.

Die beiden Arduino-Sketches für die Alarmerweiterung sind dank der RFM95-Arduino-Bibliotheken, die Sie GitHub oder der Webseite für diesen Artikel [1] entnehmen können, überraschend einfach. Lassen wir aber jetzt GreatScott! zu Wort kommen und erzählen, wie es zur Entwicklung dieses Projekts kam.



Obwohl in meiner Garage ein Alarmsystem installiert ist, das aus einer Lichtschranke, einem Stroboskop, einer Sirene und einer Controllino-SPS besteht, entschloss sich jemand vor etwa vier Monaten zu einem Einbruch. Seitdem habe ich zwar das Garagenschloss verbessert, aber das Hauptproblem ist, dass ich in meiner Wohnung von einem ausgelösten Alarm benachrichtigt werden will, auch wenn meine Garage etwa 600 m entfernt ist.

Als Lösung für mein Problem schickte mir Elektor einige Baugruppen: zwei RFM95-LoRa-Platinen mit passenden Breakout-Boards, zwei STM32-basierte Mikrocontroller-Entwicklungsboards und ein LoRa-kompatibles Gateway LGo2 Dragon.



In diesem Projekt werde ich nicht nur mit der LoRa-Technologie experimentieren, sondern auch ein kleines System mit Empfänger und Sender auf LoRa-Basis entwickeln, das eine Sirene in meiner Wohnung aktiviert, wenn der Alarm in meiner Garage ausgelöst wird.



LoRa steht für *Long Range*, und der Erfinder von LoRa, die Firma Semtech, beschreibt diese drahtlose Funktechnik als eine drahtlose Plattform mit großer Reichweite und geringem Stromverbrauch. LoRa ist zu einer der weltweit führenden Netzwerktechnologien für das Internet der Dinge (IoT) geworden. Das alles klingt recht vielversprechend für mein Projekt, da ich ja sehr wenig Daten über eine relativ große Entfernung übertragen möchte.

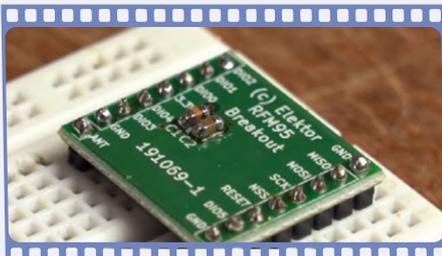
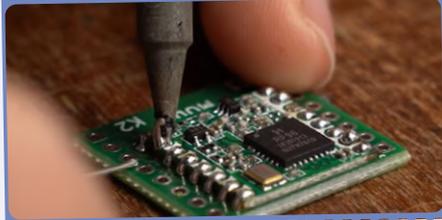
Was die Hardware betrifft, so begann ich damit, die RFM95-Platinen zusammen mit einigen Stiftheisten und Entkopplungskondensatoren auf die Breakout-Platinen zu löten.

Sehen Sie sich das Projektvideo mit der Blippar-App an

APP herunterladen

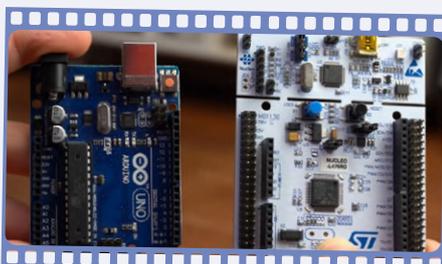


=



Laut Datenblatt handelt es sich bei den RFM95-Modulen um *Low Power LoRa-Transceiver-Module*, was bedeutet, dass sie mit geringem Energieaufwand LoRa-modulierte Daten senden und empfangen können. Dies macht sie zu Schlüsselkomponenten für mein Projekt.

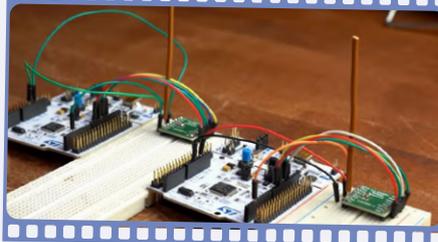
Um mit ihnen kommunizieren zu können, besitzen die Module SPI-Schnittstellen, was natürlich bedeutet, dass wir einen Mikrocontroller für die SPI-Kommunikation brauchen, und natürlich etwas Code, um die Modulregister zu schreiben und zu lesen. Für den Mikrocontroller-Teil habe ich die STM32-Nucleo-L476RG Entwicklungsboards vorgesehen, die wie Arduino Unos aussehen.



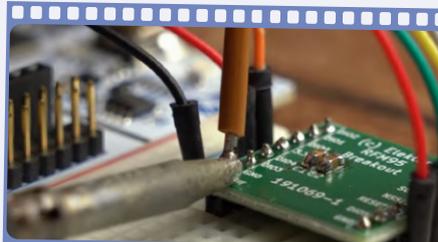
Ja, Sie könnten auch „echte“ Arduinos verwenden, um mit den RFM95-Boards zu kommunizieren, aber der Vorteil der STM32-Boards besteht darin, dass sie mit 3,3-V- statt mit 5-V-Logikpegeln (wie ein Arduino) arbeiten und das RFM95 für 3,3 V ausgelegt ist.

Die beiden Entwicklungsboards und die

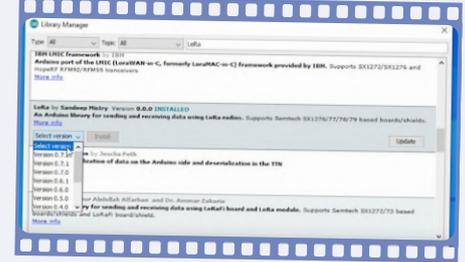
beiden LoRa-Boards werden wie in der Arduino-LoRa-Bibliothek beschrieben verbunden. Mit dieser Bibliothek wird das Programmieren zum Kinderspiel!



Natürlich brauchen wir Antennen für die LoRa-Boards, die wir leicht selber herstellen können. Nach den einfachen Berechnungen, die ich in einem Elektor-Artikel gefunden habe, brauchen wir nur ein Stück Draht mit einer Länge von etwa 82 mm. Also schnitt ich zwei solcher Drahtstücke ab und lötete sie direkt an den Antennenstift der LoRa-Platinen.

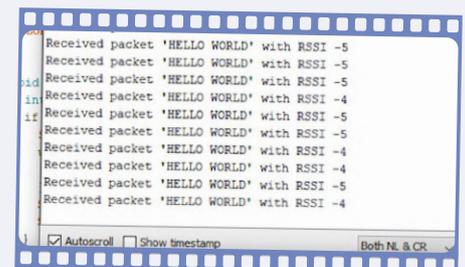


Damit war unser Hardware-Testaufbau schon abgeschlossen, und es war an der Zeit, beide Entwicklungsplatinen an meinen Computer anzuschließen. Nachdem ich die URL für die STM32-Boards-Bibliothek angegeben und im Boardsmanager installiert hatte, installierte ich auch die Arduino-LoRa-Bibliothek im Bibliotheksmanager.



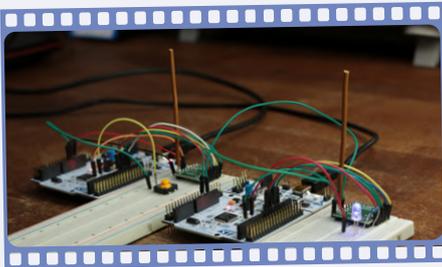
Dann schrieb ich ein wenig Code für den Sender, der ein einfaches HELLO WORLD sendet, und einen fast ebenso kleinen Code für den Empfänger, der diese Nachricht empfängt und zusammen mit der Anzeige der empfangenen Signalstärke (RSSI) auf dem seriellen Monitor ausgibt.

Nach der Auswahl der STM32-Nucleo-Boards und dem Hochladen der Codes können wir sofort sehen, dass der Empfänger die vom Sender gesendeten Nachrichten erfolgreich empfängt. Fantastisch!

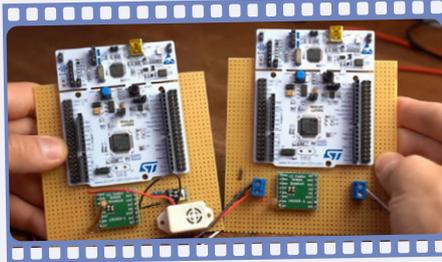
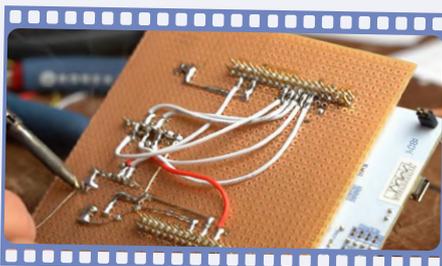




Aber ich war noch nicht fertig, denn der Sender muss einen richtigen Alarmcode aussenden, wenn er von der Alarmanlage in meiner Garage aktiviert wird, und der Empfänger muss über einen kleinen MOSFET eine Sirene einschalten, wenn er den richtigen Alarmcode empfängt. Zu Testzwecken habe ich zunächst nur einen Drucktaster am Sender als Alarmanlagenersatz und eine LED mit Strombegrenzungswiderstand als Sirenenersatz auf der Empfängerseite hinzugefügt.



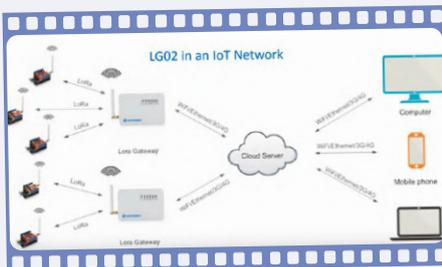
Im Sender-Sketch musste ich lediglich den Anschlusspin des Drucktasters als Eingang und eine bestimmte Alarmmeldung definieren. Beim Empfänger fiel der Sketch etwas komplizierter aus, da ich die empfangene Nachricht in einem Array speichern, mit der vordefinierten Alarmmeldung vergleichen und nur, wenn beide übereinstimmen, die LED aktivieren musste. Nachdem ich beide Sketches auf die Boards hochgeladen und einige Tests durchgeführt hatte, sah es so aus, als funktionierte alles einwandfrei. Ich konnte also richtige Schaltpläne für den Sender (**Bild 1**) als auch für den Empfänger (**Bild 2**) zeichnen. Danach zupfte ich alle Kabel und Boards von den Steckplatinen und verband die Komponenten auf zwei Stücken Experimentierplatine. Damit sollten Sender und Empfänger für den praktischen Einsatz ausreichend stabil aufgebaut sein.



Nach etwa einer Stunde Lötten waren die Lochrasterplatinen fertig. Ich wollte mich gerade mit der Senderplatine auf den Weg zu meiner Garage machen, fiel mir doch in der hintersten Ecke des Labortisches das einsame LoRa-Gateway aus dem Elektor-Paket auf, das ich völlig vergessen hatte. Ich begann mich zu fragen, ob ich es weiter vergessen könnte oder ob es doch mein Projekt gravierend aufwerten würde.



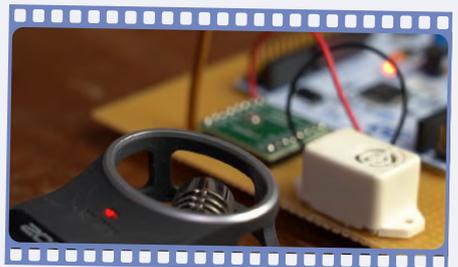
Um es kurz zu machen: Für diese einfache Anwendung lohnt es sich nicht, sich durch die Handbücher und Menüs des Gateways zu kämpfen. Das Gateway kann Daten von LoRa-Knoten empfangen, die es über Ihr Heimnetzwerk auf einen Cloud-Server hochlädt. Auf diese hochgeladenen Daten können dann andere Geräte zugreifen. Über das Gateway ist es auch möglich, Daten an andere LoRa-Knoten zu senden.



Ich würde sagen, dieses Gateway ist nützlich, wenn Sie zum Beispiel viele Temperatur- oder Feuchtigkeitsdaten für ein Gewächshaus protokollieren müssen, und Sie einen LoRaWAN-Server, MQTT- oder TCP/IP-Server verwenden. Und ich weiß, diese Begriffe wirken alle superkompliziert, wenn Sie sich - wie ich - nicht schon mit der Materie beschäftigt haben. Aber das Benutzerhandbuch gibt tatsächlich sehr gute Anleitungen

zur Anwendung dieser drei Methoden, sogar Arduino- und STM32-Sketches sind dabei. Mein Garagenalarm ist jedoch so einfach, dass er keinen Server-Upload benötigt. Außerdem möchte ich nicht gezwungen sein, eine stabile Internetverbindung zu haben, um mein System aktiv zu halten. Wie auch immer, das Gateway kann direkt zum Senden oder Empfangen von LoRa-Nachrichten verwendet werden, und es bietet sogar eine Option für angepasste Skripts, die auf spezifische LoRa-Nachrichten reagieren können. Das brachte mich tatsächlich dazu, darüber nachzudenken, ob ich es als eine Art zentralen Punkt (Hub) in mein System implementieren sollte, aber nachdem ich stundenlang versucht hatte, den korrekten Alarmcode zu senden, und nachdem ich das angepasste Linux-Shell-Skript neu geschrieben hatte, gab ich die Idee auf, weil es einfach die Dinge gewaltig verkomplizieren würde. Daher werde ich das Gateway vorerst nicht benutzen, halte es aber in Reserve, um es dann in einem anderen Projekt zu verwenden.

Um zu meiner einfachen Anwendung zurückzukehren: Ich schaltete meinen Empfänger ein und ging zur Garage, um den Sender an meine Controllino-SPS anzuschließen, bei der ich den Code nur ein wenig abändern musste. Nachdem ich den Alarm ausgelöst hatte, begann auch der Summer der Empfangsstation in meiner Wohnung zu pfeifen, was bedeutete, dass dieses Projekt ein voller Erfolg war!



Wenn Sie Reichweitenprobleme haben, können Sie versuchen, über die Einstellungen von Leistung, Spreizfaktor und Bandbreite das LoRa-System zu optimieren. Darüber erfahren Sie mehr auf der GitHub-Seite von Arduino LoRa.

```
void setup() {  
  pinMode(D4, INPUT_PULLDOWN);  
  LoRa.setTxPower(20);  
  LoRa.setSpreadingFactor(12);  
  LoRa.setSignalBandwidth(62.5E3);  
  Serial.begin(9600);  
  if (!LoRa.begin(868100000)) {  
    Serial.println("LoRa module not found");  
  }  
}
```

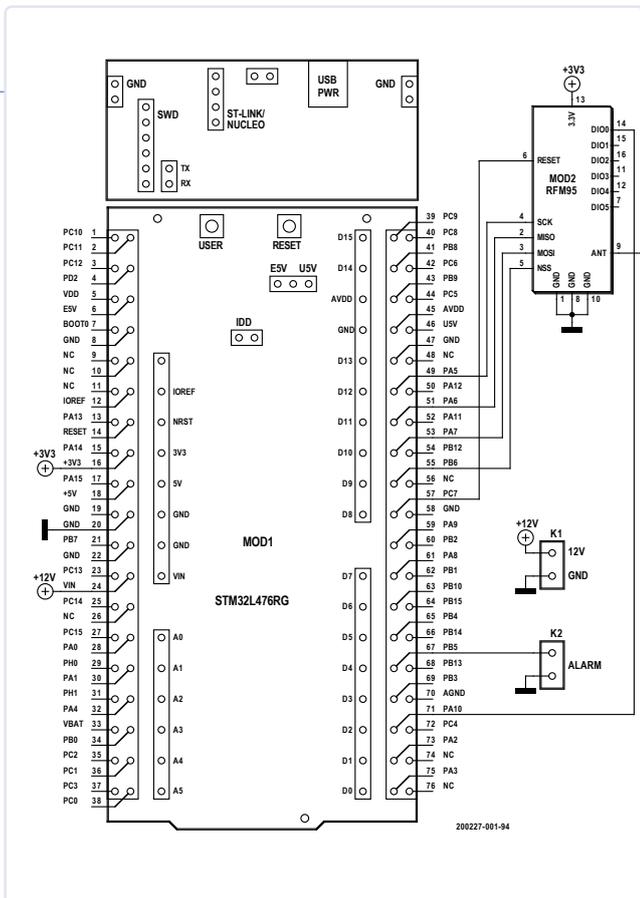


Bild 1. Verdrahtung des Senders.

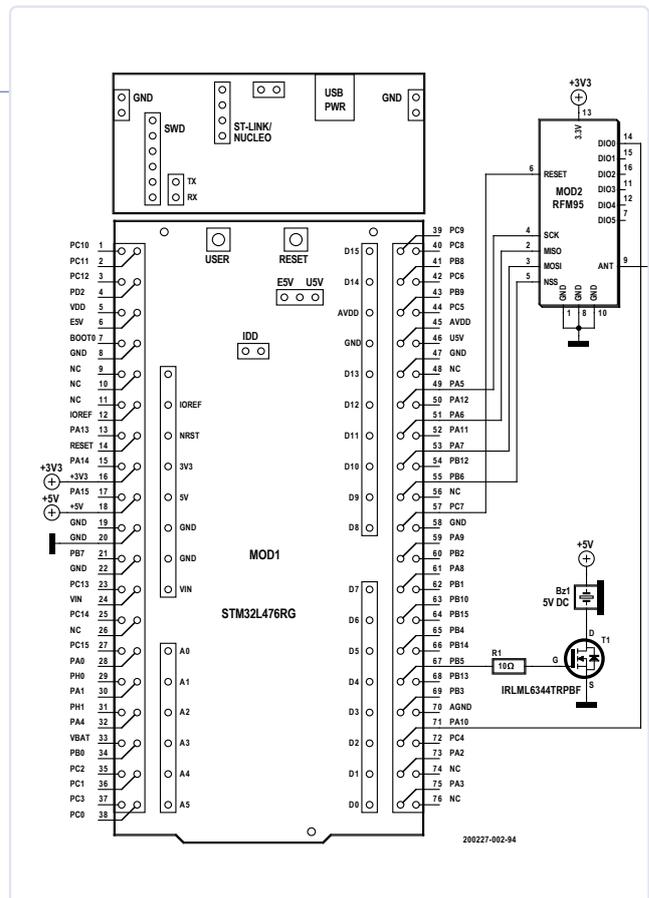


Bild 2. Verdrahtung des Empfängers.

Fazit

Mit diesem Projekt zeigt GreatScott!, dass LoRa eine einfache Lösung für einfache Kommunikation über große Entfernungen und mit geringer Geschwindigkeit sein kann. Beachten Sie, dass der FET, den er auf der Empfängerplatine verwendet, ein auf einer Breakout-Platine montierter SOT-23-Typ ist, der außergewöhnlich gute Spezifikationen für 3,3-V-Logik (max. 5 A und nur 800 mV Schwellspannung!) aufweist, aber ein TO-92-Typ wie der TN0702 würde auch funktionieren. Wenn Sie dieses Projekt aufbauen wollen, sind die wichtigsten Komponenten sämtlichst **im Elektor-Store** erhältlich. Die STM-Nucleo-Platine ist Teil eines Starterkits, kann aber auch separat gekauft werden. Der komplette Entwicklungsprozess dieses Projekts kann auch im YouTube-Video [2] betrachtet werden. ◀

200227-0



WEBLINKS

[1] **Download der LoRa/Arduino-Sketches:** www.elektormagazine.de/200227-03

[2] **GreatScott!-Video:** Transmitting an Alarm Signal with LoRa (600m)! || Improving my Garage Alarm System: <https://youtu.be/ItZwa1AdrpU>



STÜCKLISTE

Bauteile benötigt jeweils für Empfänger und Sender (also 2x):

STM32 Nucleo L476RG Board
 RFM95 Ultra LoRa Transceiver Module
 Elektor RFM95 Break-out-Board 191069-1
 C1 = 100 n, 10%, 10 V, MLCC, X7R, SMD 0805
 C2 = 10 µ, 20%, 10 V, MLCC, X5R, SMD 0805
 2 St. 1x8-polige Stiftleiste (Raster 2,54 mm)
 Draht 82 mm als Antenne

Zusätzlich für Sender:

K1, K2 = 2-polige Platinenanschlussklemme,
 Raster 5 mm

Zusätzlich für Empfänger:

R1 = 10 Ω
 T1 = IRLML6344TRPBF (optional auf
 Breakout-Board) oder TN0702 (TO-92)
 Bz1 = Summer 5 VDC



IM ELEKTOR-STORE

- **STM32 Nucleo L476RG Board (SKU 17946)**
www.elektor.de/stm32-nucleo-l476rg-board
- **Elektor STM32 Nucleo Starter Kit (SKU 19205)**
www.elektor.de/elektor-stm32-nucleo-starter-kit
- **RFM95 Break-out-Board - Leerplatine (191069-1) (SKU 19142)**
www.elektor.de/rfm95-break-out-board-bare-pcb-191069-1
- **RFM95 Ultra LoRa Transceiver Module (868/915 MHz) (SKU 18715)**
www.elektor.de/rfm95-ultra-lora-transceiver-module-868-915-mhz
- **Dragino LG02 Dual Channels LoRa IoT Gateway (SKU 18624)**
www.elektor.de/dragino-lg02-dual-channels-lora-iot-gateway

Labornetzteil-Kit Joy-iT RD6006



Von **Dr. Thomas Scherer**

Von Joy-iT gibt es jetzt einen Bausatz für ein digitales Labornetzteil mit 0-60 V und 0-6 A. Zum Zusammenbau braucht man keinen Lötcolben, sondern nur handelsübliches Werkzeug wie einen Kreuzschlitz-Schraubendreher. Was taugt das Kit?

Auch wenn „Mann“ bekanntlich nie genug Netzteile haben kann, weiß ich schon gar nicht mehr, wohin damit in meinem Elektronik-Labor. Doch da ich so positive Erfahrungen mit dem kleinen analogen Labornetzteil PeakTech 6080 A (unter [1] finden Sie das Review dazu) gemacht hatte, wollte ich mir auch mal ein größeres Labornetzteil mit digitaler Bedienung antun. Meine Wahl fiel auf das Labornetzteil RD6006 – dieses Mal von der Firma Joy-iT.

Ein Grund für diese Entscheidung war, dass es sich nicht um ein Fertigerät handelte, sondern dass das Ganze als eine Art Bausatz angeboten wird.

Bausatz

„Eine Art Bausatz“ trifft es, denn dieses Netzteil ist im Elektor-Shop als Joy-iT JT-RD6006 DC Power Supply Bundle erhältlich [2]. Gemeint ist damit nicht etwa ein „normaler Bausatz“ mit Bauteilen und Platine, die man sorgfältig bestücken muss und dann auf Funktion testen, kalibrieren und einbauen. Dieses Bundle besteht aus fertigen Modulen, die man ganz ohne Verbrauch von Lötzinn in ein Gehäuse schraubt und fertig. Die Module sind schon getestet und auch kalibriert – fast. Doch dazu später mehr.

Zunächst zu den technischen Daten:

- Netzspannung: 115/230 V (Standard)
- Ausgangsspannung: 0 – 60 V
- Ausgangsstrom: 0 – 6 A



Bild 1. Das Labornetzteil-Bundle steckt in einem imposanten schwarzen Koffer. Zum Vergleich liegt eine Elektor-Ausgabe daneben.



Bild 2. Aufgeklappt gibt der Koffer drei ebenfalls verpackte Teile preis.

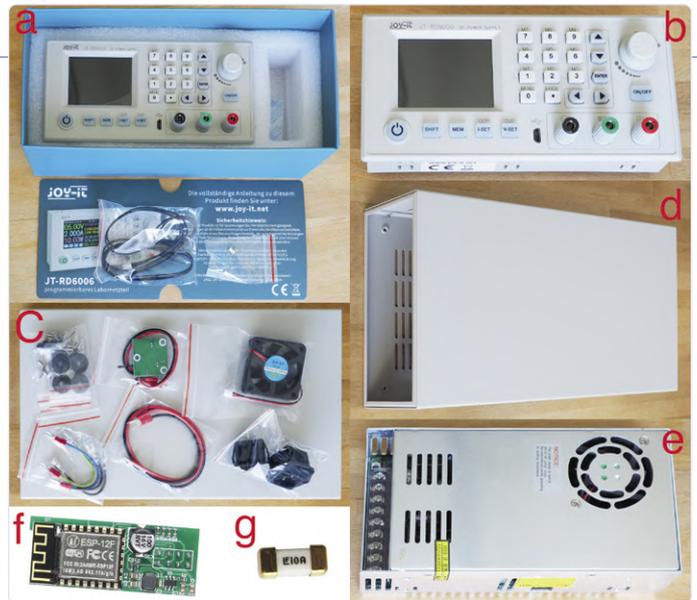


Bild 3. Inhalt des Bundles. 3a+3b: das Elektronik-Modul samt Frontblende sowie Temperatursensor und SMD-Sicherung; 3c: mechanische Kleinteile, extra Lüfter samt Steuerungsplatine, Kabel, Netzbuchse und Schalter; 3d: das Gehäuse aus Stahlblech, 3e: das 400-W-Industrienetzteil; 3f: das WLAN-Modul ESP12F; 3g: die beigelegte 10-A-SMD-Ersatz-Sicherung in voller Pracht.

- › Ausgangsleistung: max. 360 W
- › Spannungsauflösung: 10 mV
- › Stromauflösung: 1 mA
- › Ladefunktion: 0 – 9.999,99 Ah; 0 – 9.999,99 Wh
- › Restwelligkeit: 100 mV_{SS} (bei max. Last)
- › Display: 2,4“ Farb-LCD
- › Bedienung: Tastenfeld, Drehgeber, USB, WLAN
- › WLAN-Modul: ESP12F

Das Labornetzteil ist also ganz schön umfangreich ausgestattet, kann einiges mehr als mein älteres digitales 480-W-Labornetzteil chinesischer Provenienz und ist mit 175,46 € (für Elektor-Mitglieder) sogar noch etwas preiswerter. Maximal 60 V und 6 A müssten zudem für 95% aller typischen Anwendungsfälle ausreichen.

Auspackzeremonie

Das Netzteil-Bundle kommt in einem imposanten Koffer mit den Abmessungen 50x36x15 cm daher (Bild 1), und ich hatte mich schon gewundert, warum das Paket so groß war. Klappt man den Koffer auf, blickt man auf die drei Teile von Bild 2. Entpackt man die drei kleineren Pakete, kommen viele Teile zum Vorschein (Bild 3). Fast habe ich einen Moment daran gezweifelt, dass man das alles lötfrei zusammenbauen kann.

Wie man in Bild 3a sehen kann, ist keine Anleitung enthalten und leider auch kein direkter URL, sondern wie heute üblich nur ein Link auf die Webseite des Herstellers. Aber so schwierig ist das nicht. Dort in die Suche das Modell RD60006 eingegeben und schon stößt man mit einem Klick auf ein Bild auf die Bedienungsanleitung als PDF-Datei [3].

Lädt man die verlinkte Datei herunter, dann sieht man, dass es keine (Auf-)Bauanleitung ist, sondern dass es sich wirklich fast nur um die Bedienung dreht. Die Montage und Verdrahtung der vielen Teile ergibt sich zwar logisch, ist aber nicht ganz trivial.

Aufbauanleitung

Damit Sie es einfacher haben als ich, können Sie nachfolgend sehen, wie die Teile zusammengehören. An Werkzeug empfiehlt sich ein Kreuzschlitz-Schraubendreher mittlerer Größe, ein sehr kleiner „normaler“ Schraubendreher und eine kleine Spitzzange zum Festhalten der Muttern. Da wirklich kein Lötkolben erforderlich ist, habe ich das Ganze „schnell mal eben“ auf dem Küchentisch zusammengebaut. Ein Elektronik-Labor ist dazu wirklich nicht erforderlich.

Zuerst entfernt man die acht M3x5-Senkkopf-Kreuzschlitzschrauben, mit denen der Deckel des grauen Blechgehäuses befestigt ist. Dann müssen der Netzschalter und die Kaltgerätebuchse in die Ausbrüche an der Rückwand gesteckt werden (Bild 4). Der

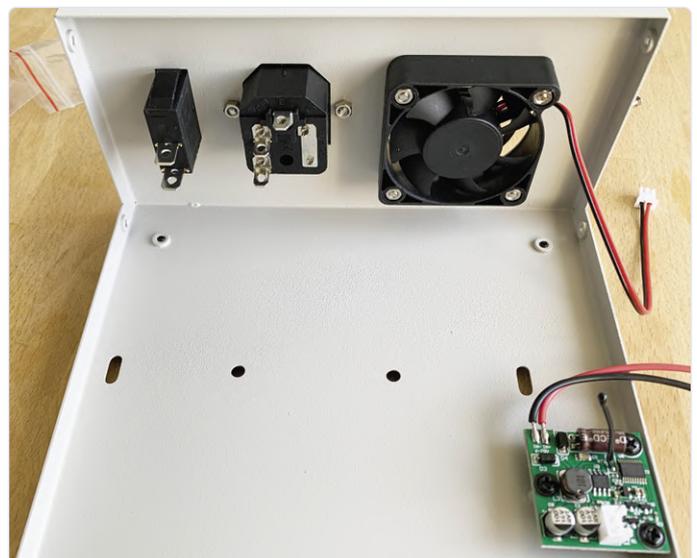


Bild 4. Befestigung von Netzschalter, Kaltgerätebuchse, Lüfter und der Platine mit der Lüftersteuerung.

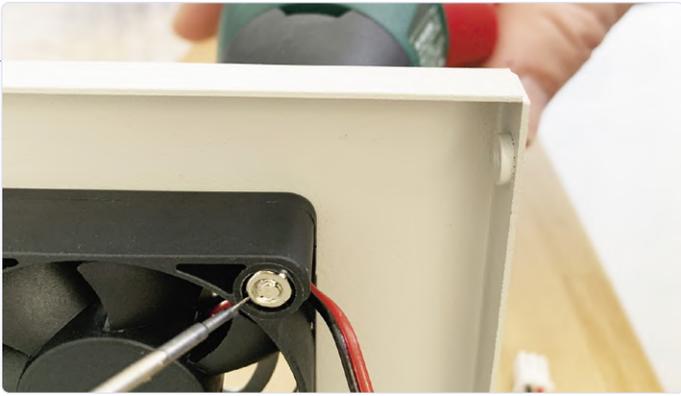


Bild 5. Ein kleiner, feiner Schraubendreher hilft beim Fixieren der Muttern beim Lüfter.

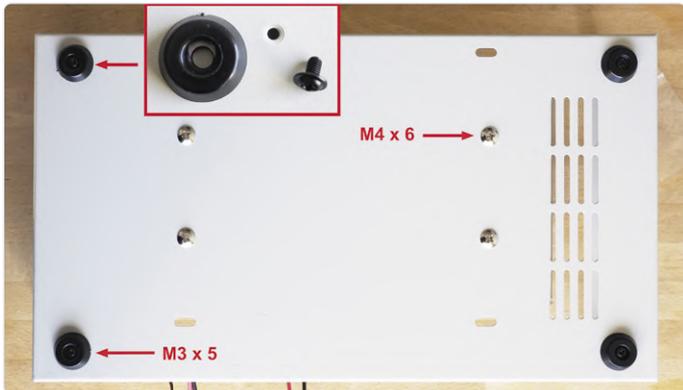


Bild 6. Die Rückseite des Metallgehäuses. Das Netzteil wird mit vier M4x6-Schrauben und die vier GummifüÙe werden mit M3x5-Schrauben befestigt. Die Netzteilanschlüsse sind hier links. Hinweis eines Lesers: Unter die Linsenkopfschrauben des Netzteils sollten Zahnringe, damit es richtigen galvanischen Kontakt zwischen dem Erdschluss der Kaltgerätebuchse und dem Gehäuse gibt.

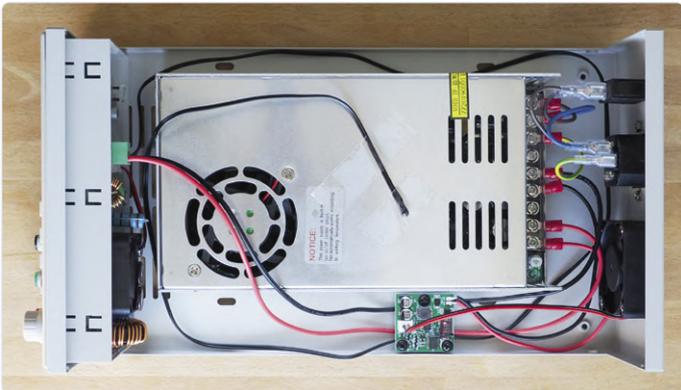
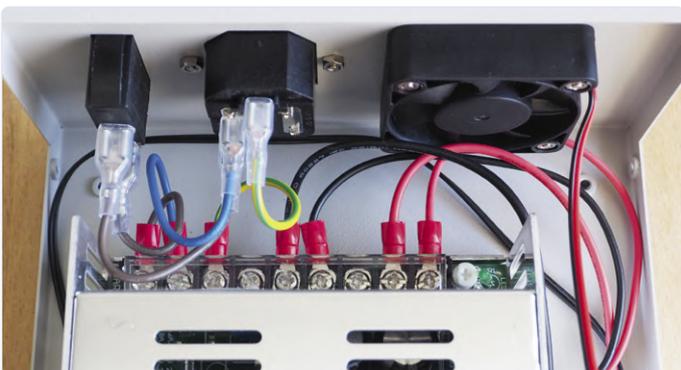


Bild 7. Alle Teile eingebaut und fertig verkabelt.



Schalter rastet ein, aber die Netzbuchse benötigt zwei M3-Schrauben samt Muttern. Hier dient die Spitzzange zum Fixieren der Muttern beim Anziehen der Schrauben. Schließlich muss noch die kleine Platine mit der Lüftersteuerung mit drei schwarzen M3x5-Schrauben befestigt werden.

Die Montage des Gehäuselüfters benötigt etwas Aufmerksamkeit: Die M3-Muttern passen zwar schön in die Versenkungen des Lüfters, doch sind diese kreisrund. In der Folge kann man die Schrauben kaum richtig festziehen, da die Muttern durchdrehen und sich ja nicht mit einer Zange fassen lassen. **Bild 5** zeigt, dass die Klinge eines sehr feinen Schraubendrehers reicht, um die Mutter zu fixieren. Der Lüfter wird so positioniert, dass sein Kabel wie abgebildet rechts liegt. Dann schraubt man das Industrienetzteil (**Bild 3e**) mit vier M4x6-Linsenkopfschrauben in das Gehäuse. Dabei ist darauf zu achten, dass die Anschlüsse des Netzteils zur Hinterseite (links) zeigen. Dann kommen die vier GummifüÙe mit je einer schwarzen M3x5-Schraube auf die Rückseite. **Bild 6** zeigt, wie das gemacht wird.

Bild 7 zeigt, wie alle Teile verkabelt werden. Besonders die Klemmen am Industrienetzteil sollten gut, aber nicht zu heftig festgezogen werden. **Bild 8** zeigt die Nahaufnahme. Natürlich darf man sich bei der Verkabelung der spannungsführenden Teile keinesfalls vertun. Alle Drähte haben eine passende Länge und die Enden sind entweder verzinkt, mit Kabelschuhen oder Steckern versehen.

Bevor man aber das Frontmodul wie in **Bild 7** zu sehen in seine Aussparung im Gehäuse schiebt, sollte man das mitgelieferte WLAN-Modul ESP12F (**Bild 3f**) hinten auf die Platine des Frontmoduls stecken. **Bild 9** zeigt das Frontmodul von hinten mit aufgestecktem WLAN-Modul. Die halb sichtbare Fassung unter dem Modul ist für eine optionale, aber nicht erforderliche Knopfzelle des Typs CR1220 vorgesehen. Die grüne Klemme „Voltage in“ kann abgezogen und mit dem längeren roten und schwarzen Kabel verschraubt und dann wieder aufgesteckt werden.

Wenn man jetzt wieder den Deckel auf das Gehäuse schraubt, sieht das fertige Labornetzteil so aus wie in **Bild 10**. Interessant ist, dass ein paar Schrauben übrig sind und auch zwei Kabelschuhe beigelegt waren, von denen ich nicht weiß, was ich damit anfangen sollte. Dafür fehlt als ausgleichende Gerechtigkeit ein Netzkabel.

Betrieb & Justage

In **Bild 11** sind vier verschiedene Betriebszustände zu sehen. Doch bevor diese beschrieben werden, noch ein paar Anmerkungen: Das farbige Display ist sehr informativ und bietet fast mehr Informationen als nötig. In dem gezeigten Standardmodus (es gibt noch einen weiteren, der Verläufe anzeigt), sieht man links die aktuellen vierstelligen Werte von Spannung (grün), Strom (blau) und Leistung (rot). Rechts wird oben die Eingangsspannung = Ausgangsspannung des Industrienetzteils unter „INPUT“ angezeigt. Darunter folgen eingestellte Spannung (U-SET) und Strom (I-SET) und dann der eingestellte Schutz gegen Überspannung (OVP) und Überstrom (OCP). In **Bild 11a** sind die aktuellen Werte alle noch null, da der Ausgang abgeschaltet ist. Betätigt man die Taste „ON/OFF“ rechts unter dem Drehgeber, dann leuchtet sie auf und der Ausgang wird eingeschaltet. In **Bild 11b** sieht man dann die aktuell anliegende und real gemessene Spannung von 5,00 V, was sich genau mit der einge-

Bild 8. Detailansicht. Bei der Verkabelung der Netzspannung führenden Teile sollte man Sorgfalt walten lassen.

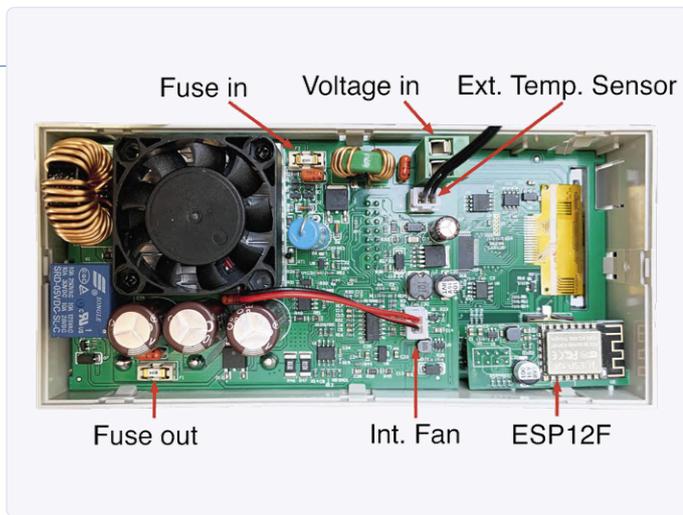


Bild 9. Das Frontmodul von hinten mit aufgestecktem ESP12F-Modul. Unter diesem Modul findet sich eine Fassung für eine (nicht benötigte) Lithium-Knopfzelle.



Bild 10. Das fertige Labornetzteil. Bei mir sind acht M3-Senkkopfschrauben und zwei Kabelschuhe übrig geblieben.

stellten Spannung deckt. Ohne Last fließt kein Strom – das Netzteil befindet sich im Modus Konstantspannung (CV ganz unten in weiß). Meine Messung ergab genau 4,996 V, was auf einen vernachlässigbaren Fehler von -0,08 % hinausläuft. Ergo ist das Netzteil sehr gut kalibriert worden. Nicht nur das überzeugt. Mein älteres 480-W-Netzteil hat nur eine Anzeige von Spannung und Strom. Hier sieht man nun eventuelle Differenzen, und das ist gut so. In **Bild 11c** habe ich versucht, die maximal mögliche Spannung einzustellen. Möglich waren aber nur 59,18 V statt 60,00 V bei U-SET. Warum? Das lässt sich leicht aufklären: Die Elektronik des

Labornetzteils ist ja zweigeteilt: Das Industrienetzteil liefert hier 60,18 V (bei voller Last einige 10 mV weniger, wegen des Widerstands von Sicherung und Kabeln). Daraus macht das Frontmodul dann einstellbare 0 V bis eben nicht ganz die Eingangsspannung. Der Buck-Regler kommt da nämlich nicht ganz ran. Also habe ich mit dem in Bild 7 gut zu sehenden kleinen, weißen Poti die Eingangsspannung hochgedreht. Bei 61,5 V kann dann die Ausgangsspannung wirklich bis 60,00 V einstellen. Damit es unter Last keine Probleme gibt, habe ich noch eine kleine Reserve vorgesehen. **Bild 11d** zeigt, dass eine

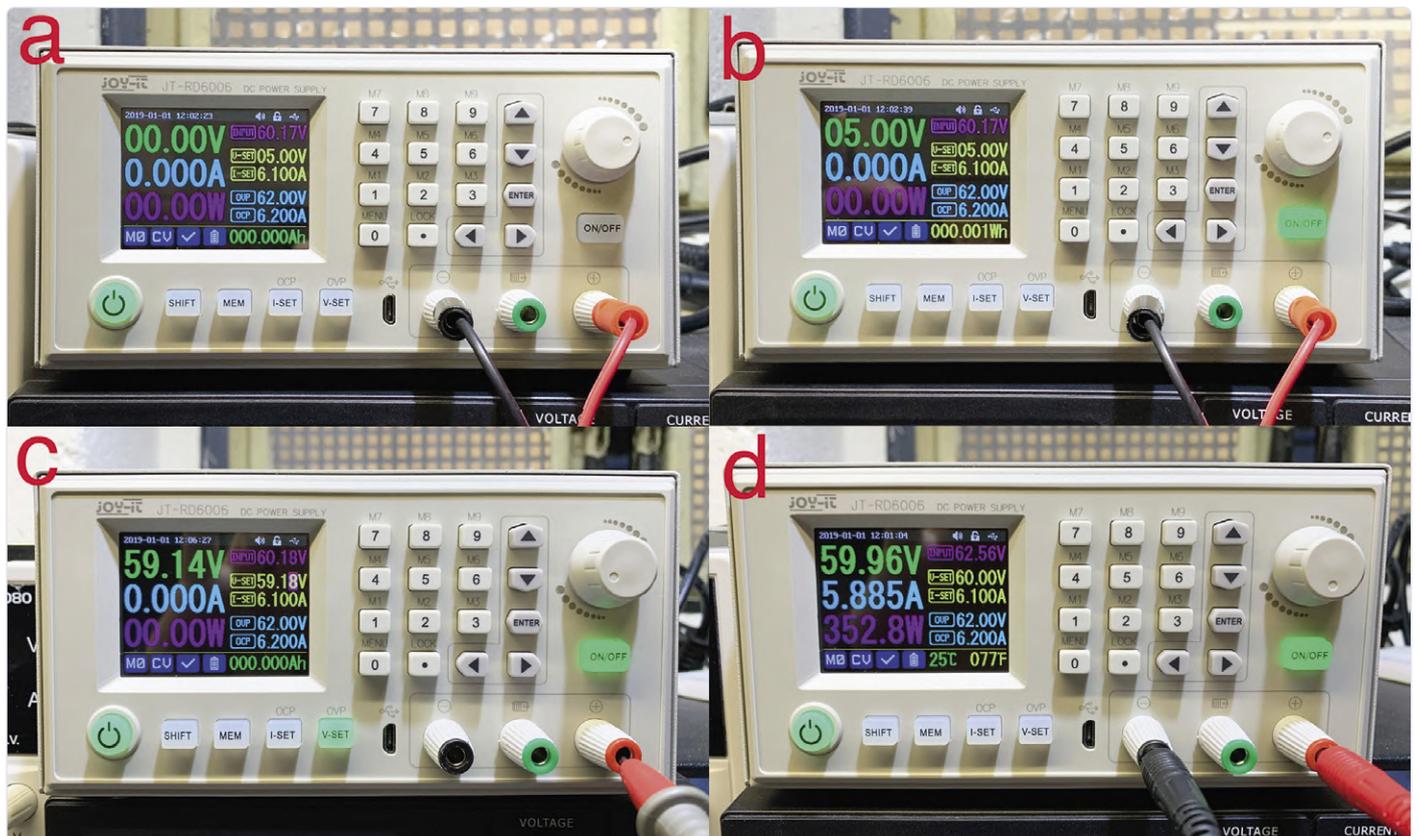


Bild 11. Display bei vier verschiedenen Modi. 11a: Ausgang abgeschaltet; 11b: Ausgang mit 5 V aktiv; 11c: Ausgang max. 59,14 V bei Eingang = 60,13 V; 11d: Ausgang 59,96 V bei Eingang = 62,56 V.

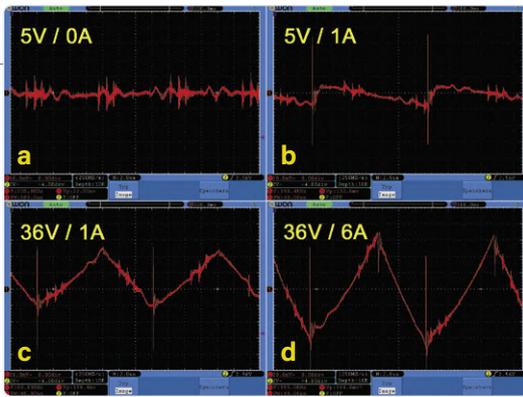


Bild 12. Screenshots der Störpegel der Ausgangsspannung unter vier verschiedenen Lastbedingungen.

Eingangsspannung von 62,56 V auch für volle Last ausreicht. Bei 5,885 A liegen am Ausgang dann echte 59,96 V an und die Kabel zu meinen Lastwiderständen wurden warm. Mein Multimeter meint, es wären 59,94 V gewesen – ein völlig vernachlässigbarer Fehler. Mit etwas Nachjustierung läuft das Labornetzteil also recht perfekt. Auch die Lüfter waren kaum zu hören und die interne Temperatur hatte nach 15 Minuten bei 353 W Ausgangsleistung nur 29° erreicht.

Störpegel

Es ist klar, dass ein Schaltnetzteil keine so saubere Ausgangsspannung liefert wie ein analoges Pendant. Also habe ich mein Oszilloskop an die Ausgangsklemmen angeschlossen und geschaut, was für Störsignale so auftreten. **Bild 12** zeigt die Störpegel bei vier verschiedenen Bedingungen. Zunächst: Ein messbarer 50- oder 100-Hz-Brumm war nicht nachweisbar – soweit also schon mal sehr gut. Bei langsamer Horizontalablenkung war allerdings schon ohne Last eine Art statistisches Rauschen zu sehen. Also habe ich gesucht und bei einer Ablenkung von 2 μs /Teilstrich konnte man dann periodische Störsignale mit einer Grundfrequenz von etwa 115 kHz sehen – die eingebaute Frequenzmessung meines Oszilloskops vertut sich nicht nur hier aufgrund der komplexen Kurvenform der Störsignale. Der Pegel lag bei **Bild 12a** (ohne Last) bei etwa 22 mV_{SS}.

Bei einer schwachen Last von 5 W (**Bild 12 b**) erniedrigt sich die Grundfrequenz auf rund 70 kHz und es treten periodische Spikes mit einer Amplitude von 130 mV_{SS} auf. Ein kleiner Tiefpass am Ausgang hätte die energiearmen Spikes beseitigt und die Störampplitude auf ca. 35 mV_{SS} gesenkt. Bei der moderaten Last von 36 W (**Bild 12c**) bleiben die Frequenz und auch die Spikes, dafür zeigt sich eine schöne Dreieckswelle mit einer Amplitude von etwa 70 mV_{SS}. In **Bild 12d** kann man bei der hohen Last von etwa 200 W sehen, dass sich die Amplitude des Dreiecksignals auf rund 130 mV_{SS} fast verdoppelt. Das sind zwar etwas mehr als die vom Hersteller angegebenen maximal 100 mV_{SS}, das Ganze bleibt aber für ein digitales Labornetzteil in durchaus akzeptablem Rahmen.

Fazit & Sonstiges

Nicht probiert habe ich, das Netzteil über USB oder per WLAN fernzusteuern. Die Angaben im Handbuch waren für mich eher kryptisch und – Hand aufs Herz – ich brauche sowas auch nicht. Ich stelle Spannungen und Ströme immer noch gerne von Hand direkt am Gerät ein. Ich habe also die sonstigen eingebaute Luxusfunktionen schlicht ignoriert, da ich sie nicht brauche. Übrigens kann dieses Labornetzteil angeblich auch Akkus mit definierten Werten laden. Dafür gibt es sogar für den Pluspol eine eigene grüne Ausgangsbuchse. Hier wird der Strom abgeschaltet, wenn er unter 10 mA fällt, damit ein eventuell angeschlossener Akku nicht überladen werden kann. Mir hat sich aber nicht erschlossen, wieso man dafür eine extra Buchse braucht. Und um zu verstehen, wie die Akkuladung funktioniert, braucht es wohl die Fähigkeiten von Alan Turing, um diese Passagen des Handbuchs zu entschlüsseln. Auch die englische Version des Handbuchs macht nicht viel klüger, denn auch sie wurde wohl aus dem Chinesischen übersetzt.

Was die Möglichkeiten angeht, gefällt mir dieses Labornetzteil besser als das, was ich schon seit Jahren nutze. Insbesondere das Display ist klasse, weil es Soll- und Istwerte zugleich anzeigt. Ich mag es auch, dass man (bis zu zehn) Presets für Spannung und Strom definieren und nutzen kann. Auch die Einstellung der Werte via Drehgeber (schnell & intuitiv) oder per Tasten (genau) ist eine tolle Sache. Schön finde ich, dass Spannung und Strom sehr genau kalibriert sind. Die Störpegel am Ausgang sind zwar nicht so gut wie bei einem analogen Labornetzteil, aber in der Praxis (außer für die Versorgung empfindlicher HF-Teile) völlig ausreichend niedrig. Nach dem Einschalten bootet der „integrierte Computer“ aka Mikrocontroller extrem fix: Das Netzteil ist innerhalb einer Sekunde betriebsbereit, wenn man die Anzeige des Logos abschaltet. Das ist super, denn mein altes Netzteil strapaziert mit fast 15 s meine Geduld ganz ordentlich. Das PreisLeistungsverhältnis ist gut und im Defektfall kann man sogar leicht einzelne Teile austauschen.

Nun zu den Schattenseiten: Dass kein Netzkabel beiliegt, geschenkt. Was stört ist die Qualität des Handbuchs. Auch wenn ich die für mich relevanten Infos finde, hätte das ein richtiger Elektroniker verfassen sollen. Mich persönlich ärgert das aber nicht besonders, da ich ja die „verschlüsselten“ Funktionen nicht brauche. ◀

200130-01



IM ELEKTOR-STORE

> Joy-iT RD6006

www.elektor.de/joy-it-jt-rd6006-dc-power-supply-bundle

WEBLINKS

- [1] **Review:** www.elektormagazine.de/news/reviewlabornetzteilpeaktech6080a
- [2] **Joy-iT RD6006:** www.elektor.de/joy-it-jt-rd6006-dc-power-supply-bundle
- [3] **Bedienungsanleitung:** <https://bit.ly/2WBK2Rt>

Interface-Board GreatFET One

Von **Tam Hanna**

Wer nach einem schnellen Weg sucht, um einfache MSR-Aufgaben mit einem PC zu erledigen, findet hier eine technisch interessante und für Python-Programmierer bequem zu bedienende Lösung.

Die Weiterentwicklung der Computertechnik - Stichworte wie Mehrkernprozessoren oder LCD-Bildschirme - brachten dem Nutzer im Allgemeinen nur Vorteile. Das gilt leider nicht für den Elektroniker - während die Interaktion mit Gameport, serieller Schnittstelle und Druckerinterface vergleichsweise einfach war, setzen Arbeiten mit der USB-Schnittstelle ohne Unterstützung

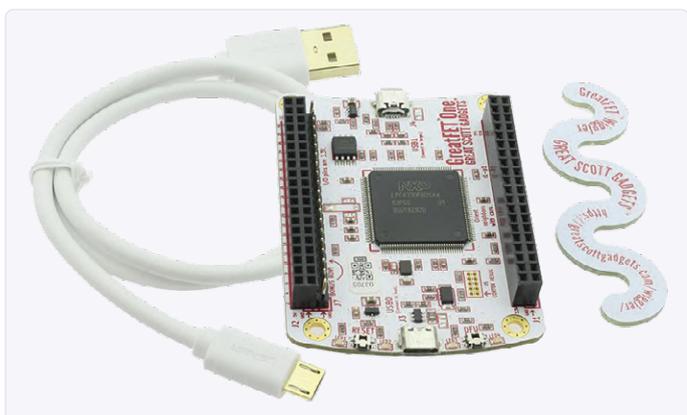
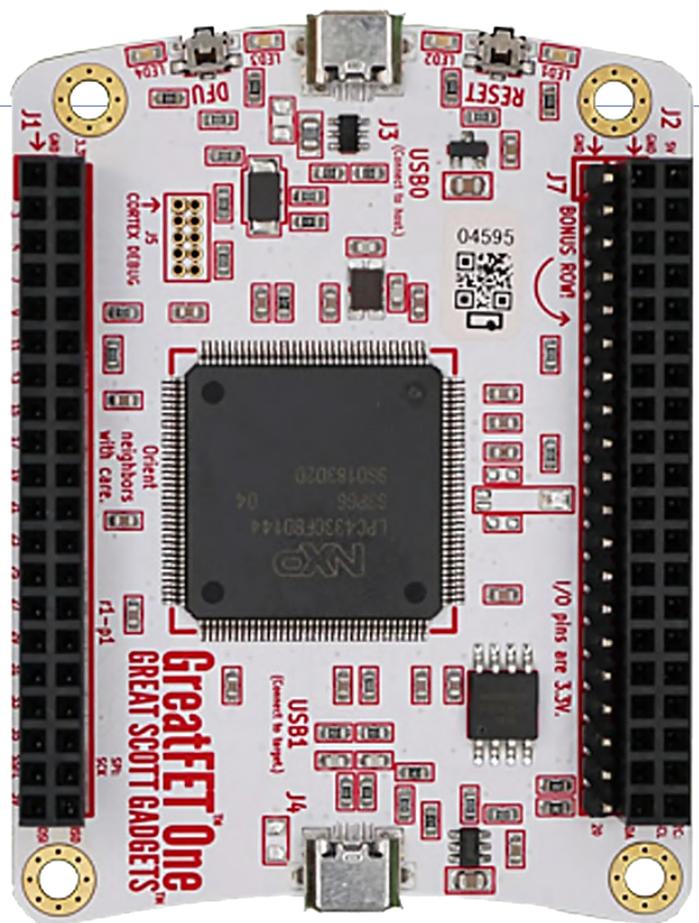


Bild 1. Die Auslieferung des GreatFET One erfolgt schlüsselfertig.



dedizierter Chips gute Programmierkenntnisse voraus. Great Scott Gadgets bietet mit dem GreatFET One nun ein Entwicklungsboard an, das die Rückkehr in die „gute alte Zeit“ ermöglichen soll.

Was bekomme ich?

Der GreatFET One ist durch die Bank als quelloffene Hardware ausgelegt - eine Beschreibung der Software findet sich in einem Wiki [1], während die auf Kicad basierenden Hardware-Designs ebenfalls auf Github zum Download bereitstehen [2]. Wer den GreatFET One kauft, bekommt das in **Bild 1** gezeigte Paket - neben der (betriebsbereiten und mit allen Headern ausgestatteten) Platine bekommen Sie auch ein USB-Kabel und ein als Wiggler (englisch für Rüttler) bezeichnetes Werkzeug zum Trennen des Boards von aufgesteckten Erweiterungen.

Dokumentation zur als *LibGreat* bezeichneten Haupt-Interaktions-API findet man übrigens unter [3].

Schnelle Einrichtung!

Wenn ein Entwickler schnell ein Programm zusammenschustern möchte, dann greift er heutzutage zur Python-Programmierung. Der Autor hat die in den folgenden Schritten erfolgende Schnell-Demonstration unter Ubuntu 18.04 LTS durchgeführt. Wer stattdessen unter Windows arbeiten möchte, findet Installationsanleitungen im Netz [4].

In jedem Fall beginnt die Installation der Bibliothek mit dem pip3-Paketmanager:

```
tamhan@TAMHAN18:~$ pip3 install --upgrade --user
greatfet
```

Python bringt seit längerer Zeit eine eigene Paketverwaltung mit, über die wir das GreatFET-Paket samt seinen teilweise sogar nativen Abhängigkeiten laden. Wichtig ist, dass Sie auf jeden Fall pip3 verwenden müssen - der für Python 2.X vorgesehene pip

```
tamhan@TAMHAN18:~$ greatfet info
Found a GreatFET One!
Board ID: 0
Firmware version: 2018.12.1
Part ID: a0000a30584f06
Serial number: 000057cc67e6303a6757
```

Bild 2. Der GreatFET One ist sichtbar.

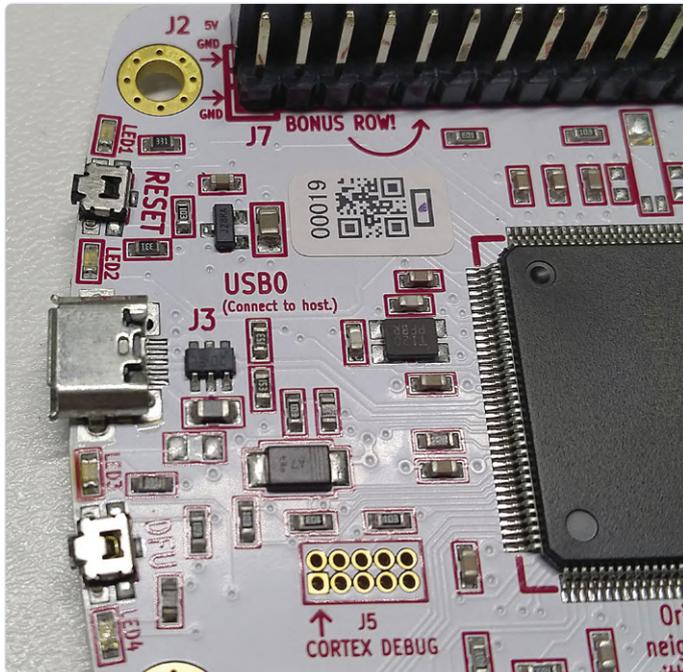


Bild 3. Klare Beschriftung ist Trumpf: Der für die PC-Kommunikation vorgesehene USB-Port ist ebenfalls klar ersichtlich.

funktioniert nicht. Im nächsten Schritt überprüfen wir, ob unser User-Konto schon Mitglied der PlugDev-Gruppe ist:

```
tamhan@TAMHAN18:~$ groups tamhan
tamhan : tamhan adm dialout cdrom sudo dip plugdev
        lpadmin sambashare kvm pico
```

Vor dem Verbinden des Prozessrechners mit dem Computer geben wir noch folgende Kommandos ein, um die udev-Regeln zu aktualisieren:

```
tamhan@TAMHAN18:~$ sudo wget https://raw.githubusercontent.com/greatscottgadgets/greatfet/master/host/misc/54-greatfet.rules -O /etc/udev/rules.d/54-greatfet.rules
tamhan@TAMHAN18:~$ sudo udevadm control -reload-rules
```

Lohn der Mühen ist, dass Sie einen angeschlossenen Prozessrechner - wie in **Bild 2** gezeigt - durch Eingabe des Kommandos `greatfet info` erkennen können.

Der GreatFET One ist mit Sicherheit kein Produkt, das Schlomo Normalentwickler jeden Tag kauft. Schon aus diesem Grund liegen die PCBs einige Zeit beim Distributor auf Halde, weshalb sich eine Firmware-Aktualisierung durch das folgende Kommando auf jeden Fall empfiehlt:

```
tamhan@TAMHAN18:~$ gf fw --auto
```

Der Autor hat in den folgenden Schritten mit der Firmware-Version v2020.1.2 gearbeitet.

Kombinatorische Prozessrechnersysteme leiden im Allgemeinen darunter, dass die Bandbreite zwischen den einzelnen Modulen eng beschränkt ist. Zum Test der Möglichkeiten möchte der Autor im ersten Schritt eine charakteristische Wellenform ausgeben. Hierzu benötigen Sie eine .py-Datei, die man durchaus bequem in Visual Studio Code editieren kann:

```
tamhan@TAMHAN18:~/greatfetspace$ code worker.py
tamhan@TAMHAN18:~/greatfetspace$ python3 worker.py
```

Ein netter Nebenaspekt der Microsoft-IDE ist, dass das Terminal sofort nach der Aktivierung retourniert - der Aufruf des Python-3-Interpreters kann also an derselben Stelle erfolgen. Im nächsten Schritt können wir nach dem folgenden Schema ein GreatFET-Objekt erzeugen und ein Pin-Objekt anlegen:

```
from greatfet import GreatFET

gf = GreatFET()
pin = gf.gpio.get_pin('J1_P4')
pin.set_direction(gf.gpio.DIRECTION_OUT)
```

Great Scott umgeht die von Raspberry Pi und Co. bekannten Probleme bei der Korrelation zwischen logischem und physikalischem Pin insofern, als jedes der Header mit einem eindeutigen Label ausgestattet ist. Der hier verwendete String steht dann beispielsweise für den physikalischen vierten Pin des Headers J1 (siehe **Bild 3**). Darauf folgt auch schon eine Endlosschleife, die die Wellenform ausgibt:

```
while 1 == 1:
    pin.write(True)
    pin.write(False)
    pin.write(True)
    pin.write(False)
```

Die **Bilder 4 und 5** informieren uns darüber, dass die beiden Wellentäler ungefähr gleich lang sind und die gesamte Affäre weder besonders schnell noch besonders frequenzstabil abläuft.

Beachten Sie beim Experimentieren übrigens, dass das Board ein reines 3,3-V-Gerät ist: Wer 5 V anschließt, brät den Controller.

Experimente

In der guten alten Zeit war das Einsammeln analoger Werte und die Ausgabe derselben durchaus haarig. Der GreatFET One bringt von Haus aus einen Analog-Digital-Konverter mit, die sich am einfachsten durch einen der vielen Kommandozeilen-Helfer aktivieren lässt. Von Haus aus wird übrigens die am Pin J2_P5 anliegende Spannung gemessen:

```
tamhan@TAMHAN18:~/greatfetspace$ greatfet adc
3.18076171875V
```

Zur Demonstration der immensen Leistungsfähigkeit der Kombination aus Python und Hardware möchte ich an dieser Stelle sowohl die NumPy-Bibliothek als auch Matplotlib laden - fehlen die beiden Bibliotheken, so müssen Sie sie von Hand nachinstallieren. Im

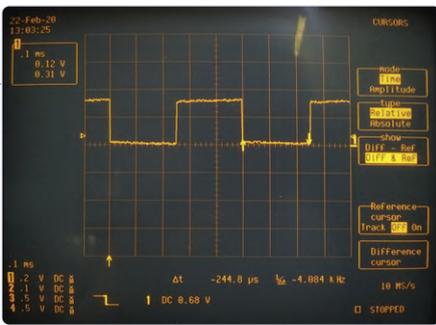


Bild 4. Das Testsignal aus Sicht eines Oszilloskops ...

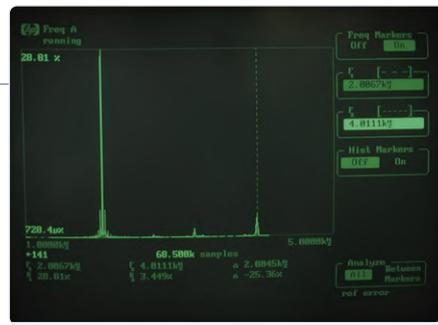


Bild 5. ... und aus Sicht eines Modulationsdomänenanalysators.

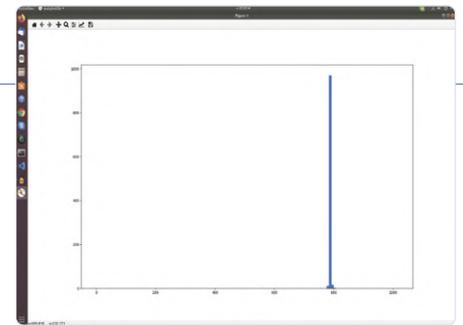


Bild 6. Histogramme mit minimalem Aufwand.

nächsten Schritt greife ich auf meine bewährte Spannungsreferenz zurück, der ich - in der Teedose ist es toastig warm - etwas Zeit zum Aufwärmen spendiert habe. Währenddessen beschäftige ich mich mit der Erzeugung des folgenden Programms. Im ersten Schritt lade ich wie gewohnt einige Bibliotheken:

```
import numpy as np
from matplotlib import pyplot as plt
from greatfet import GreatFET
```

Die implementierte ADC-Lesefunktion ist eine komplexe Angelegenheit. Im Interesse der Bequemlichkeit erzeuge ich das NumPy-Array auf die brutale Art - angemerkt sei, dass die Verwendung der API nicht sonderlich effizient ist:

```
gf = GreatFET()

store = np.array(gf.adc.read_samples(1))
i = 0
while i < 1000:
    store = np.append(store, gf.adc.read_samples(1))
    i=i+1
```

An dieser Stelle stehen zwei verschiedene Vorgehensweisen zur Verfügung. Methode Nummer eins ist die Erzeugung eines Bin-Range, den ich danach an die NumPy-Funktion `histogram` übergebe:

```
binrange = range(0,1024,8)
hist, bins = np.histogram(store, binrange)
```

Lohn der Mühen dieser Variante ist die Returnierung des Arrays, in dem sich die in den jeweiligen `bins` befindlichen Werte-Häufigkeiten abernten lassen. Die Übergabe von `bins` ist dann nur eine Kopie des Parameters `binrange`. Möchte man die Diagramme nur direkt anzeigen, so kann man die

Histogramm-Anzeige auch nach dem folgenden Schema direkt an Matplotlib auslagern:

```
plt.hist(store, bins)
plt.show()
```

Lohn der Mühen ist jedenfalls das in **Bild 6** gezeigte Histogramm - bitte ignorieren Sie die EMI-bezogenen Unsauberkeiten. Zu guter Letzt sei noch darauf hingewiesen, dass der GreatFET auch einen Pattern-Generator und sogar eine Funktion zur Exportierung von Sigrok-Log-Informationen bietet. Beide lassen sich nach dem folgenden Schema über Kommandozeilen-Befehle aktivieren:

```
tamhan@TAMHAN18:~/greatfetspace$ greatfet logic -p
out.sr -f 2M -n 4
```

Die genaue Arbeitsgeschwindigkeit ist dabei allerdings stark von der verwendeten Hardware abhängig - unter [5] findet sich eine interessante Diskussion der Thematik.

Fazit

Der GreatFET ist trotz seiner etwas „durch den Wind verteilten“ Dokumentation und der im Test eher mittelmäßigen Echtzeit-Performance ein liebenswertes Stück Hardware. Wenn Sie schon immer in die „gute alte Zeit“ der COM-Schnittstelle zurückkehren wollten, finden Sie hier einen schnellen und bequemen Weg, um PC oder Laptop für MSR-Aufgaben fit zu machen. ◀

200124-01

IM ELEKTOR-STORE

➤ **GreatFET One:** www.elektor.de/greatfet-one

WEBLINKS

- [1] **Software:** <https://github.com/greatscottgadgets/greatfet/wiki>
- [2] **Hardware-Design:** <https://github.com/greatfet-hardware/azalea>
- [3] **LibGreat Doku:** <https://github.com/greatscottgadgets/libgreat>
- [4] **Installation unter Windows:** <https://greatscottgadgets.github.io/greatfet-tutorials/windows.html>
- [5] **Geschwindigkeit:** <https://github.com/greatscottgadgets/greatfet/issues/286>

Von der Pike auf gelernt - XXL

Diesmal: Altes aus der Elektor-Ideenkiste

Zusammengestellt von **Eric Bogers**

Seit einem halben Jahr veröffentlichen wir jede Woche auf unserer Website und im Newsletter eine kleine Schaltung, die ohne spezielle, exotische Bauteile und mit einem normalen Lötcolben auf einer Lochrasterplatine aufgebaut werden kann. Manchmal ist es eine Schaltung aus der tiefsten Laborkiste, manchmal eine aktuelle Einsendung eines Lesers oder eine Idee aus unserem Labor. Diese Schaltungen werden auf der Website diskutiert - manchmal kontrovers - und genau das ist unsere Absicht: Wir wollen Diskussionen, Experimente und eigene Entwicklungen der Leser anregen! Hier ist eine Auswahl aus den ersten 26 Episoden.

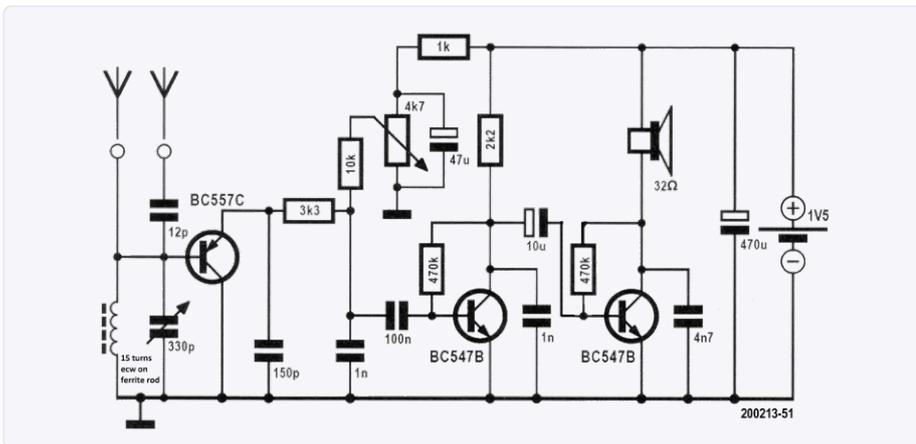


Bild 1. Das Diagramm dieses Empfängers ist nicht kompliziert und der Nachbau nicht sehr kritisch.



Idee: Burkhard Kainka (Deutschland)
Viele Elektronikeinsteiger schrecken davor zurück, einen Radioempfänger selbst zu bauen – zu hohe Frequenzen, gruselige Spulen... Dass es aber leicht möglich ist, mit einer Handvoll Standardbauteilen einen Empfänger zu bauen, beweist diese Schaltung.

Kurzwellenempfänger

Der hier vorgestellte dreistufige Kurzwellenempfänger (**Bild 1**) verfügt über einen Drehkondensator zur Senderwahl und eine zusätzliche Einstellung für die Rückkopplung. Im Prinzip ist die erste Stufe ein Oszillator. Der Arbeitspunkt dieses Oszillators bzw. seine Verstärkung kann jedoch in einem weiten Bereich eingestellt werden. Der Trick am Audion-Prinzip ist, dass die Verstärkung genau so eingestellt ist, dass die Verluste des

Schwingkreises gerade ausgeglichen werden. Das Ganze fängt also gerade nicht an, von selbst zu schwingen. An diesem Punkt zeigt der Empfänger seine größte Empfindlichkeit und die beste Selektivität.

Die PNP-Oszillatorstufe (in Kollektorschaltung) fungiert nicht nur als Audion, sondern demoduliert gleichzeitig das (amplitudenmodulierte) HF-Signal. Die beiden nachfolgenden NF-Stufen liefern genügend NF-Leistung, so dass direkt ein kleiner Lautsprecher angesteuert werden kann. Ein gebräuchlicheres 8-Ω-Exemplar ist leiser.

Ob das Kurzwellen-Audion auf einen Punkt eingestellt werden kann, an dem noch keine Schwingungen auftreten, hängt auch von der Dämpfung der Eingangsschaltung durch die angeschlossene Antenne ab. Daher sind hier zwei Antennenanschlüsse vorgesehen.

Bei Anschluss über den 12-pF-Kondensator erfolgt eine lose Kopplung und die Dämpfung ist gering. Die direkte Verbindung hingegen ist besonders für kurze Antennen geeignet (eine längere Antenne strahlt nämlich mehr HF-Energie ab, was den Schwingkreis dämpft). Bei richtiger Einstellung ist der Empfänger extrem empfindlich. In der Anfangszeit der Funktechnik war dieses Schaltungsprinzip für Empfänger normal. Selbst schwache Sender in Entfernungen von mehreren tausend Kilometern konnten damit empfangen werden.



Idee: Elex-team

Zugegeben: Dämmerungsschalter mit viel Schnickschnack kann man für wenig Geld kaufen. Aber macht es nicht viel mehr Spaß, so etwas selbst zu bauen – mit sage und schreibe zehn Bauteilen?

Einfacher Dämmerungsschalter

Ein Dämmerungsschalter ist ein (elektronischer) Schalter, der eine Lampe o.ä. einschaltet, wenn es dunkel wird - unter anderem, um potenzielle Einbrecher abzuschrecken, indem er suggeriert, dass jemand zu Hause ist. Heutzutage kann man diese Dinge zwar aus Fernost für ein paar Euro kaufen, aber der Selbstbau macht Spaß und ist überhaupt nicht schwierig, wie die Schaltung in **Bild 2** beweist.

Natürlich braucht es ein lichtempfindliches Bauelement, denn irgendwoher muss die Elektronik ja wissen, wie hell oder dunkel es gerade ist und dass die Dämmerung beginnt. So kann sie entscheiden, ob es dämmt. In

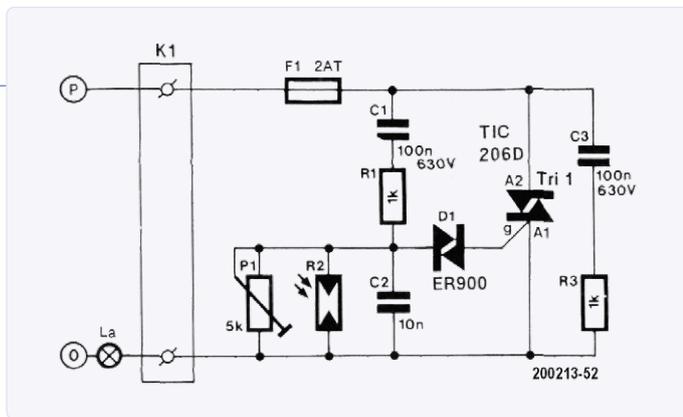


Bild 2. Das exotischste Bauteil in dieser Schaltung ist zweifellos der Diac.

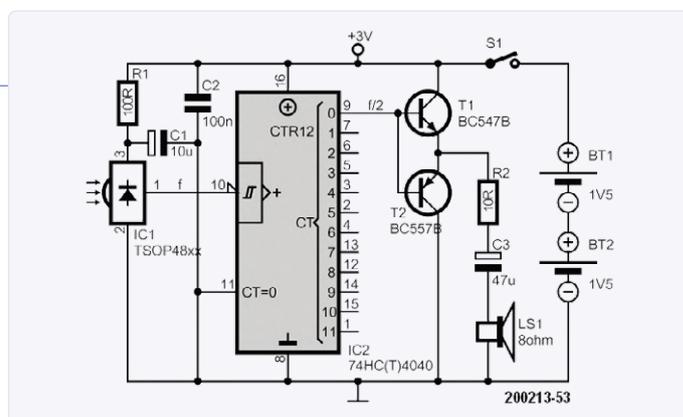


Bild 3. Um die Signale der IR-Fernbedienung hörbar zu machen, müssen Sie nur deren Frequenz herunterteilen.

der Schaltung ist hierfür der LDR (Light Dependent Resistor) R2 zuständig. Sein Widerstand ist um so niedriger, je stärker er beleuchtet wird. Damit ist der zentrale Aspekt der Schaltung auch schon klar: Die Kombination aus C1, R1, C2, R2 und P1 produziert eine lichtabhängige Wechsellspannung. Die Kombination ist dabei voll an 230 V angeschlossen. Wenn es draußen dunkler wird (was angeblich seit Jahrmilliarden einmal am Tag geschieht), steigt der Widerstand von R2 und damit auch die Spannung an der Verbindung von R1 und C2.

Bei einer bestimmten Dunkelheit (einstellbar mit P1) erreicht dort die Wechsellspannung einen Wert von ca. 30 VSS, dann schaltet Diac D1 durch. Dadurch wird C2 über D1 und das Gate von Triac Tri1 entladen. Mit anderen Worten: Der Triac erhält einen Zündimpuls und leitet so lange, bis sein (geringer) Haltestrom unterschritten wird. Die Lampe leuchtet, zunächst sanft und mit steigender Dunkelheit immer heller.

Die gesamte Schaltung ist direkt an die Netzspannung angeschlossen. Der Triac erhält also bei 50 Hz Wechsellspannung 100 Mal pro Sekunde einen Zündimpuls. Er leitet dann immer so lange, bis der Strom durch die Lampe beim nächsten Nulldurchgang nahezu Null wird. Bei beginnender Dämmerung wird der Triac relativ spät in diesen 10 ms gezündet und die Lampe glimmt eher. Bei Dunkelheit kommt der Zündimpuls früher und die Lampe leuchtet heller. Der Diac hat den Vorteil, dass er ab einer bestimmten Spannung sehr niederohmig wird und erst wieder sperrt, wenn sein (sehr geringer) Haltestrom unterschritten wird. Das ist der Fall, wenn C2 entladen ist und sicher bei jedem Nulldurchgang der Wechsellspannung. Der Stromimpuls von C2 durch D1 ins Gate von Tri1 ist somit auf jeden Fall groß genug, um Tri1 sicher definiert zu zünden.

Aufbau

Nochmals: Die gesamte Schaltung liegt an der Netzspannung! Die Montage muss

daher mit der notwendigen Sorgfalt und unter Beachtung der geltenden Sicherheitsbestimmungen für Netzspannung berührungssicher erfolgen. Am besten ist es, die Schaltung in einer berührungssicheren Verteilerbox zu installieren. Lassen Sie im Zweifel jemanden einen Blick darauf werfen, der sich mit sowas wirklich auskennt!



Idee: Ton Giesberts (Elektor-Labor)
Haben Sie sich schon mal gefragt, wie eine IR-Fernbedienung eigentlich klingt? Mit dieser kleinen Schaltung können Sie das herausfinden!

Akustischer IR-Fernbedienungs-Tester

In der Schaltung in **Bild 3** wird das Ausgangssignal eines Standard-IR-Empfängers IC1 so modifiziert, dass man damit einen Lautsprecher ansteuern kann. Der hier verwendete IR-Empfänger von Vishay ist für verschiedene Modulationsfrequenzen von 30 bis 56 kHz erhältlich. Viele IR-Fernbedienungen verwenden das sogenannte RC5-Protokoll bei einer Frequenz von 36 kHz. In unserem Prototyp kam der Typ TSOP4836 zum Einsatz. Schon aus der Typennummer lässt sich schließen, dass dieser Empfänger für 36 kHz gedacht ist. Er funktioniert (mit geringerer Empfindlichkeit) aber auch mit benachbarten Frequenzen, wenn sie nicht zu sehr abweichen.

Natürlich ist das Signal am Ausgang des IR-Empfängers mit den avisierten 36 kHz deutlich zu hochfrequent, um es hören zu können. Deshalb wird es zunächst durch einen Teiler-IC (IC2) gejagt und auf wahrnehmbare Tonhöhen gebracht. Je nach Geschmack kann man auch das Signal von einem der anderen Ausgänge verwenden. In der Schaltung wurde mit Pin 9 der Ausgang „Q0“ verwendet. Aber Achtung: Manche Hersteller bezeichnen Pin 9 auch als „Q1“, was eigentlich korrekter

ist, wenn man den Divisor nach der Formel 2^N berechnet, denn hier liegt die halbierte Frequenz an. Ältere Zeitgenossen teilen also besser durch 4, 8 oder gleich 16.

Mit dem geteilten Signal wird eine sehr einfache Gegentakt-Endstufe aus T1 und T2 angesteuert. Sie verstärkt den Strom ausreichend für einen kleinen Lautsprecher. Die Schaltung wird mit 3 V aus zwei AA-Batterien versorgt. Der Stromverbrauch beträgt maximal 13 mA bei hörbarem Ton und etwa 0,66 mA in Ruhe. R1 und C1 entkoppeln die Versorgung für den IR-Empfänger, um Störungen der Versorgungsspannung durch die Ausgangsstufe zu vermeiden.

Mit dieser Schaltung, die leicht auf einem Steckbrett oder einer Lochrasterplatte aufgebaut werden kann, kann man nicht nur prüfen, ob eine IR-Fernbedienung noch funktioniert, sondern mit etwas Übung auch verschiedene Marken und Modelle erkennen ;-). Zugegeben – der praktische Nutzen dieses Testers ist relativ gering, aber die Schaltung eignet sich hervorragend für sonntäglichen Bastelspaß und taugt als Basis für weitergehende Entwicklungen.



Idee: Elex-Team
In realen und virtuellen Tierhandlungen werden Heerscharen von Ultraschall-Hundepfeifen verkauft. Aber mit ein paar Bauelementen können Sie so etwas leicht selbst bauen!

Elektronische Hundepfeife

Ultraschall-Hundepfeifen werden in den Situationen eingesetzt, in denen Herrchen (oder Frauchen) dem Vierbeiner Kommandos geben will, die nur der Hund hören soll – zum Beispiel zum Training. Ultraschall eignet sich allerdings auch zu anderen Zwecken. Bei entsprechender Lautstärke kann so eine Hundepfeife auch dazu verwendet werden, aggressive Hunde

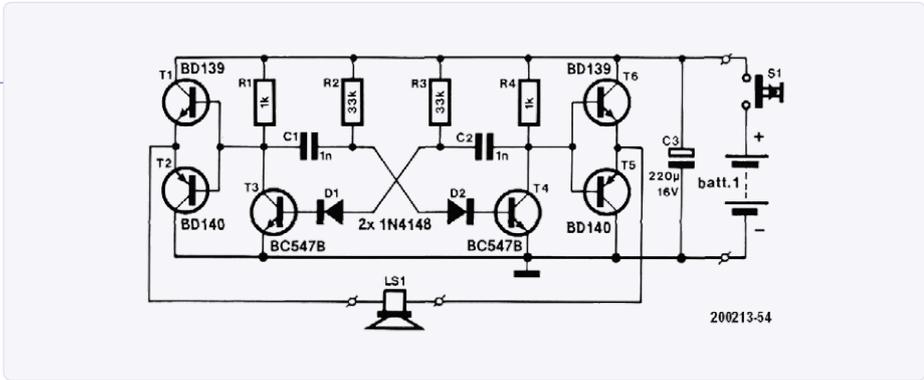


Bild 4. Eigentlich ist dies nichts anderes als ein diskret aufgebauter astabiler Multivibrator mit Leistungsstufen.

(oder welche mit Appetit auf das Steak auf dem Grill im Garten) zu verjagen. Der für Menschen nicht wahrnehmbare Ton ist aufgrund seiner Lautstärke für Hunde so unangenehm, dass sie seine Nähe meiden und das Weite suchen. Nicht nett, aber wirksam.

In der simplen Schaltung in **Bild 4** bilden T3 und T4 zusammen einen klassischen astabilen Multivibrator (d.h. freilaufenden Rechteck-Oszillator). Seine Frequenz beträgt mit den angegebenen Bauteilwerten etwa 21 kHz. Aufgrund des vollständig symmetrischen Aufbaus erzeugt der Multivibrator eine saubere, symmetrische Rechteckwelle. D1 und D2 verhindern, dass die Schaltung neben der gewünschten Ultraschall-Rechteckwelle hörbares Rauschen erzeugt.

Das 21-kHz-Rechtecksignal wird durch zwei Gegentakt-Ausgangsstufen (T1/T2 und T5/T6) stromverstärkt, sodass der Piezo-Hochtöner LS1 mit entsprechender Leistung angesteuert werden kann. Ursprünglich wurde der Hochtöner KSN1001A von Monacor verwendet, doch wenn Sie so etwas im Internet suchen, werden Sie viele moderne Piezos finden.

Die Schaltung ist für die Stromversorgung aus einer 9-V-Blockbatterie ausgelegt. Zum Schluss noch eine Warnung: Aufgrund des hohen Geräuschpegels sollte diese Schaltung nicht in der Nähe von empfindlichen Hundehören verwendet werden - außer natürlich, wenn Sie angegriffen werden. Und natürlich können Sie mit der hier beschriebenen Schaltung nach Herzenslust experimentieren. Mit anderen Bauteilwerten und einem normalen Lautsprecher können Sie auch eine elektronische Hupe oder was auch immer realisieren!



Idee: Burkhard Kainka (Deutschland)
Früher, als Halbleiter noch nicht allgegenwärtig waren, benutzte man eine Glimmlampe und einen Kondensator, um einen Sägezahn zu generieren...

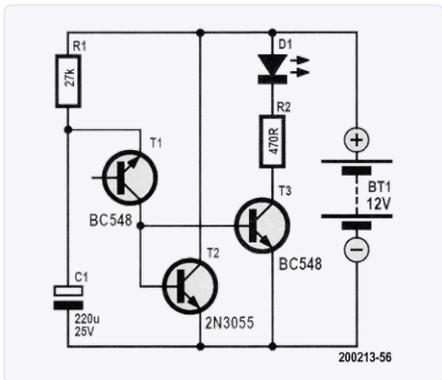


Bild 6. Der Akku-Refresher besteht aus bemerkenswert wenigen Bauteilen.

Akkus frisch halten

Die untenstehende, super-einfache Schaltung zeigt, wie man mit Hilfe einer Glimmlampe und eines Kondensators ein Sägezahnsignal generieren kann. Die Schaltung ist auch als Relaxationsoszillator oder Kippschwinger bekannt. Weniger bekannt ist, dass dies auch mit einem „normalen“ NPN-Transistor (in der Schaltung ein BC547) funktioniert. Eine detaillierte Erläuterung der Hintergründe würde hier zu weit führen. Der Hinweis, dass die Schal-

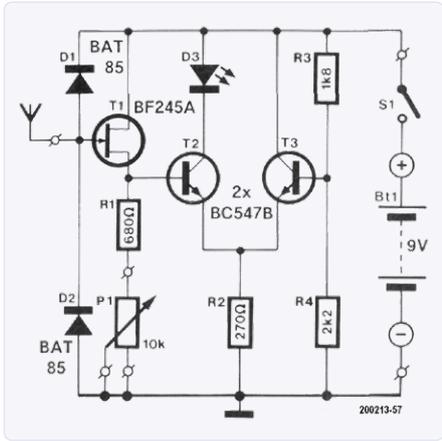


Bild 7. Nicht so raffiniert und vielseitig wie ein moderner Leitungssucher, dafür kann dieses Exemplar im Handumdrehen gebaut werden.

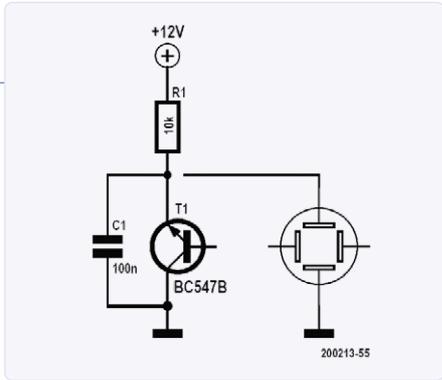


Bild 5. Der ursprüngliche Relaxationsoszillator und sein Transistoräquivalent.

tung in **Bild 5** funktioniert, und ein Verweis auf die Originalveröffentlichung in Elektor 7/8 2011 (Seite 106) müssen genügen.

Themawechsel: Blei-Gel-Akkus haben die unangenehme Eigenschaft, dass sie hochohmig werden, wenn sie längere Zeit nicht benutzt werden. Und obwohl es manchmal möglich ist, einen solchen toten Akku wieder halbwegs zum Leben zu erwecken, ist das mühsam, langwierig und klappt nicht immer gut. Besser ist es daher, es gar nicht so weit kommen zu lassen.

Eine bewährte Vorsichtsmaßnahme ist das kontinuierliche Laden des Akkus mit sehr kurzen, aber recht kräftigen Stromimpulsen. Dies geschieht mit der Schaltung in **Bild 6**. Der aufmerksame Leser wird darin den NPN-Kippschwinger aus der vorigen Schaltung wiedererkennen.

Der Oszillator um T1 steuert die Basis des Leistungstransistors T2 durch, der den Akku (bei den angegebenen Werten) alle zwei Sekunden mit einem 1-A-Impuls von etwa 2 ms beaufschlagt. Dies entspricht einem durchschnittlichen Strom von etwa 1 mA, was in der Größenordnung der Selbstentladung eines Blei-Gel-Akkus liegt. Es besteht also kein Grund zur Sorge.

Die Teilschaltung aus T3, D1 und R2 dient als grober, visueller Indikator des Ladezustands: Je seltener die LED aufleuchtet, desto niedriger ist die Akkuspannung.



Idee: Elex-Team
Natürlich haben Sie schon mal Löcher in die Wand gebohrt, um etwas aufzuhängen - aber wie ist das mit den Stromleitungen, die so schön in der Wand hinter der Tapete lauern? Es gibt Spaßigeres, als diese anzubohren...

Leitungsfinder

Abgesehen von der Gefahr eines Stromschlags und dem fast sicheren, vollen Kurzschluss durch das Anbohren eines

Stromkabels, machen die Folgen eines solchen Malheurs keinen Spaß. Auf jeden Fall muss das beschädigte Kabel ersetzt (oder repariert) werden. Das geht noch, wenn die Kabel sauber in Rohren verlegt sind. Aber genau das ist in Deutschland oft nicht der Fall, manchmal ist so eine Leitung einfach unter Putz verlegt. Es kommt auch vor, dass es trotz Rohren heftig klemmt, wenn man ein neues Kabel einziehen will. Oft mündet das Anbohren einer Leitung daher in einer kleinen Katastrophe.

Vorbeugen ist besser als heilen, weshalb der schlaue Heimwerker einen Leitungsfinder benutzt. Aber was, wenn man an einem verregneten Sonntagnachmittag zur Bohrmaschine greift, wenn der Baumarkt geschlossen ist und man eben nicht schnell einen fertigen Leitungsfinder kaufen kann? Man kann so ein Gerät mit wenig Aufwand ganz unkompliziert und schnell selbst bauen!

Schaltung

Drei Transistoren und etwas Drumherum - das ist schon alles in der Schaltung in **Bild 7**. An das Gate des FET T1 ist eine Art Antenne angeschlossen (ein etwa 20 cm langes Stück Kupferdraht). Der Draht fängt das das allgegenwärtige „Netzbrummen“ auf. Die Antenne wird durch den sehr hochohmigen FET praktisch nicht belastet. Die Schottky-Dioden D1 und D2 schützen das Gate vor zu hohen Spannungen.

Das so gepufferte Antennensignal steht dann an der Source von T1 zur Verfügung. Der Gleichspannungspegel kann mit P1 eingestellt werden. Von hier gelangt das Signal an die Basis von T2, der als eine Art Komparator fungiert: Er vergleicht die Spannung an der Basis mit der Spannung am Emittor. Und letztere wird von T3 generiert. Bei der hier

angegebenen Dimensionierung beträgt die Spannung an R2 etwa 4,25 V. T2 wird leitend, wenn die Spannung an seiner Basis 4,95 V übersteigt. Dann leuchtet auch die LED D3. Mit P1 wird die Schaltung so eingestellt, dass die LED im Ruhezustand einfach nicht leuchtet. Die Schaltung ist so einfach, dass man sie fix auf einem Stückchen Lochrasterplatte realisiert hat. Wenn man die Schaltung etwas stromsparender machen will, kann man den LED-Strom durch $R2 = 820 \Omega$ auf 5 mA reduzieren (sie leuchtet mit modernen LEDs immer noch hell genug) und für R3 und R4 zehnfach höhere Werte wählen.



Idee: Kees van der Geer (Niederlande)

Schlichtweg das Gießen vergessen - Ruhe in Frieden, du teurer Bonsai. Welch verschwendete Mühe... Aber mit diesem „Dürrealarm“ ersparen Sie sich solche Katastrophen!

Dürrealarm

Die Grundidee der Schaltung in **Bild 8** ist die Einfachheit selbst: Wenn die Erde im Topf trocken wird (und die Pflanze daher Durst hat), steigt ihr elektrischer Widerstand. Man misst diesen Widerstand irgendwie, und sobald er zu groß (die Erde zu trocken) wird, ertönt der Alarm. Die Schaltung besteht im Wesentlichen aus den beiden Komparatoren eines TLC3702. Der linke Komparator um IC1A ist als Oszillator bzw. AMV (Astabiler Multi-Vibrator) geschaltet. Bei den gegebenen Bauteilwerten beträgt die Periodendauer etwa 2 s. Über C2 und C3 gelangen kurze positive und negative Impulse an den invertierenden Eingang des zweiten Komparators, der als MMV (Monostabiler Multi-Vibrator) geschal-

tet ist. C2 und der Widerstand aus R5 plus Erdwiderstand fungieren als Differenzierer. Die negativen Impulse werden durch D1 daran gehindert, wirklich negativ zu werden. Über den mehr oder weniger feuchten Boden wird ein größerer oder kleinerer Teil der Impulsamplitude über beide Elektroden nach Masse abgeleitet. Anders ausgedrückt: Je trockener die Erde im Topf wird, desto weniger wird der Impuls abgeschwächt. Die positiven Impulse haben keinen Effekt, aber durch die Wechsellspannung (via C2) wird keine elektrochemische Korrosion an den Elektroden verursacht, was sie haltbarer macht. Wenn der negative Impuls am invertierenden Eingang des rechten Komparators groß genug ist, wird der MMV ausgelöst und die LED leuchtet für etwa 100 ms auf - parallel dazu erzeugt der Summer einen deutlichen Piepton. Es piept rhythmisch so lange, bis die durstige Pflanze gegossen wird.

An sich ist diese Schaltung nicht bemerkenswert. Aber sie ist extrem energieeffizient. Selbst wenn LED und Summer kontinuierlich alle paar Sekunden aktiviert werden, hält die Batterie (ein 9-V-Block) monatelang. Dies ist ein Verdienst des verwendeten ICs und der besonders hohen Widerstände.

Anmerkung: Im Moment wird an einem Update dieser Schaltung gearbeitet; in der nächsten Ausgabe kommen wir darauf zurück!



Idee: Friedrich Lischek (Deutschland)

Vor einer Zeit haben wir im Rahmen dieser kleinen Schaltungen ein „energieeffizientes Relais“ vorgestellt. Hier eine super-du-per-einfache Variation dieses Themas.

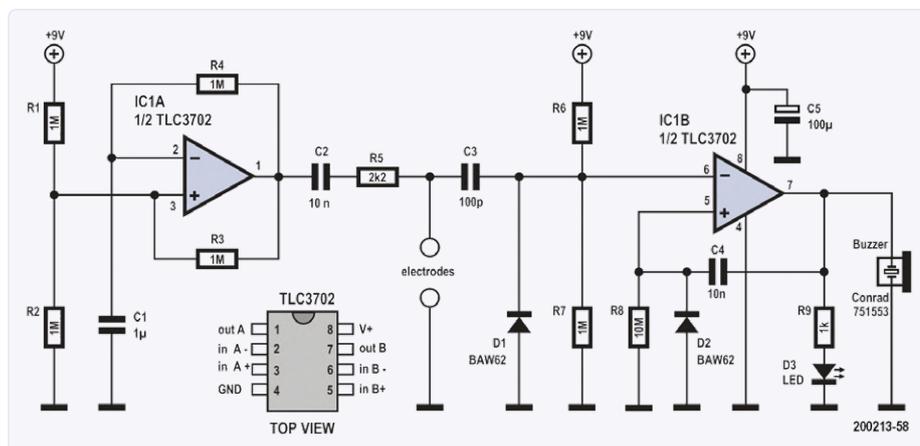


Bild 8. Zwei Komparatoren bilden das Herzstück des Dürre-Alarms.

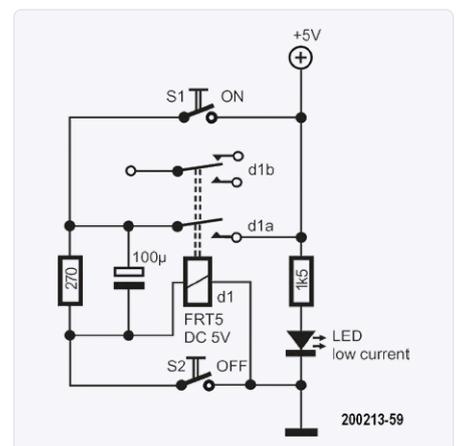


Bild 9. Transgender: Das ursprünglich monostabile Relais arbeitet hier bistabil.

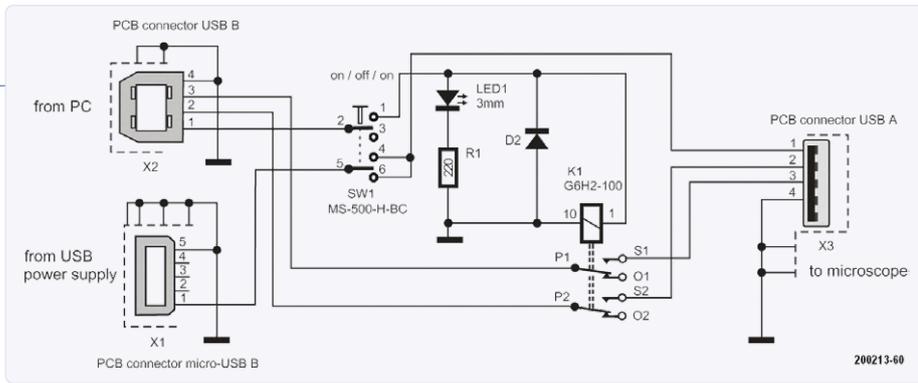


Bild 10. Achtung: Der Schalter muss vom Typ On/Off/On sein!

Energiesparendes Relais

Der große Nachteil des in dieser ersten Folge beschriebenen energiesparenden Relais war, dass Taster mit einem unterbrechenden Kontakt – also „Öffner“ – benötigt wurden. Diese sind manchmal schwer oder nicht in der gewünschten Ausführung zu beschaffen, und man hat sie normalerweise auch nicht in einer Schublade herumliegen.

Die folgende kleine, aber intelligente Schaltung in **Bild 9** verwendet normale Taster mit einem „Schließer“ als Kontakt und ein gewöhnliches 5-V-Standardrelais mit zwei umschaltenden Kontakten aka „Wechsler“. Das Schöne an dieser Schaltung ist, dass das gewöhnliche, monostabile Relais (der Schließer bleibt nur so lange geschlossen wie die Spule erregt ist; ohne Spulenstrom kehrt der Kontakt in den Ruhezustand zurück) hier als bistabiles Relais arbeitet: Nach dem Drücken von „ON“ bleibt der Schließer geschlossen, bis „OFF“ gedrückt wird.

Dieses Verhalten wird durch Nutzung einer der beiden Relais-Kontakte zur Selbsthaltung erreicht. Es dürfte nicht schwierig sein, die Funktionsweise der Schaltung zu ergünden. Nach dem Drücken von S1 wird das Relais aktiviert und seine Kontakte „umgelegt“ – durch den haltenden Relaiskontakt bleibt die Spule über den Widerstand 270 Ω erregt genug, um den Kontakt umgelegt zu halten. Dank des Elkos wird der Einschaltstrom beim Einschalten für das Relais kurzfristig erhöht, so dass es sicher anzieht. Der Haltestrom ist niedriger als der fürs Anziehen nötige Wert, daher reicht der durch den Widerstand reduzierte Strom zum Halten aus, nachdem es einmal angezogen ist. Mit S2 schaltet man das Relais wieder ab, da so seine Spule kurzgeschlossen wird. Das Relais fällt ab und der Ruhezustand ist wieder hergestellt.

Die optionale LED zeigt lediglich an, dass die Schaltung betriebsbereit ist. Ein schönes Detail: Das Relais wird nicht aus Zufall aktiviert, wenn die Stromversorgung einge-

schaltet wird. Man muss also wirklich auf S1 drücken, um es zu aktivieren. Und sollte jemand auf die Idee kommen, beide Knöpfe gleichzeitig zu drücken: Kein Problem, da passiert nichts, denn S2 hat Vorrang.



Idee: Dr. Martin Oppermann (Deutschland)

Das Andonstar ADSM302 USB-Mikroskop ist ein beliebtes Hilfsmittel für fremelige Arbeiten im Elektroniklabor (genau wie sein großer Bruder AD407). Es hat allerdings ein paar nun ja, un schöne Eigenheiten, die in mit der hier beschriebenen kleinen Schaltung auf elegante Art und Weise behoben werden können.

Verbesserungen für Andonstar ADSM302

Das ADSM302 kann als eigenständiges Mikroskop oder in Kombination mit einem PC verwendet werden. Im ersten Fall sieht man Live-Bilder auf dem Display des Mikroskops selbst – im anderen Fall wird das Bild auf dem PC-Bildschirm angezeigt.

Im PC-Betrieb wird das Mikroskop über den USB-Port dieses PCs mit Strom versorgt, und hier tritt das erste Problem auf: Ein USB-2.0-Port liefert maximal 500 mA, was eigentlich nicht ausreicht (selbst wenn die Beleuchtung des Mikroskops nicht eingeschaltet ist). Ein USB-3.0-Anschluss ist eigentlich dringend erforderlich.

Um zwischen dem Stand-Alone-Betrieb und dem PC-Betrieb umzuschalten, muss das Verbindungskabel zwischen Mikroskop und PC jedes Mal ein- oder ausgesteckt werden – dies ist der Lebensdauer von Steckern und Anschlüssen nicht gerade förderlich. Die

kleine Zusatz-Schaltung in **Bild 10** löst beide Probleme auf einen Schlag. Es ist ein Mikro-USB-Bus vorgesehen, so dass ein USB-Netzteil zur Versorgung des Mikroskops verwendet werden kann.

Mit Hilfe des Ein-/Aus Schalters kann zwischen Stand-Alone- und PC-Betrieb umgeschaltet werden. Die beiden Schaltkontakte des Miniaturrelais aktivieren die USB-Datenleitungen vom PC zum Mikroskop oder trennen sie ab. Die Art des Schalters ist wichtig: die Schaltfolge ist „break before make“. Durch kurzes Trennen der Stromversorgung des Mikroskops bei jedem Schalten wird es zurückgesetzt und die Elektronik kann die richtige Betriebsart (Stand-Alone oder PC) erkennen. Die LED leuchtet im PC-Modus.

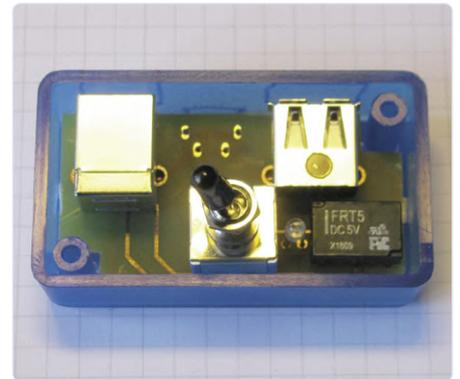


Bild 11. Der Autor hat für die Schaltung eine Platine entworfen, deren Layout von der Projektseite zu diesem Artikel heruntergeladen werden kann.

Die Schaltung kann sogar auf einer Experimentierplatine aufgebaut werden. Für die USB-Buchsen können Sie auf Wunsch einige billige USB-Kabel „opfern“. Oder Sie können (wie der Autor) eine „richtige“ Platine entwerfen, USB-Buchsen für Leiterplattenmontage verwenden und das Ganze in einer ordentlichen kleinen Box (**Bild 11**) montieren. Die Eagle-Layout-Dateien für die Platine finden Sie im Download zu diesem Artikel [1].

200213-02

WEBLINK

[1] **Projektseite:** www.elektormagazine.de/200213-02

Des armen Mannes externe 2,4-GHz-WLAN-Antenne

Von **Hans-Henrik Skovgaard**

Sie könnten zu Recht argumentieren: „Warum soll ich meine eigene WLAN-Antenne bauen, wenn es doch all diese billigen Fertiganntenen gibt? Ist der Bau nicht ziemlich kompliziert?“

Bevor wir beginnen, muss ich zunächst betonen, dass mein Design auf dem basiert, was in meinem Lab gerade verfügbar war, so dass es in vielerlei Hinsicht sicher verbesserungswürdig sein könnte. Und ich sollte auch betonen, dass sehr genau und sorgfältig gearbeitet werden muss, um einem Vergleich mit einer käuflichen (Keramik- oder anderen) Antenne standzuhalten.

Zunächst habe ich bei Aliexpress.com für weniger als 10 € 20 Kabel mit der Bezeichnung „IPX IPEX u.fl Female 1.13mm Connector Cable Single-head Adapter Connector 15cm IPX 1.13 Cable“ erstanden (**Bild 1**). Wenn Sie eine Antenne entwerfen und bauen, müssen Sie zunächst die Wellenlänge des Signals kennen, das von der Antenne abgestrahlt und/oder aufgenommen werden soll. Die Wellenlänge λ eines Funksignals kann berechnet werden mit der Formel: $\lambda = c / f$ [m].

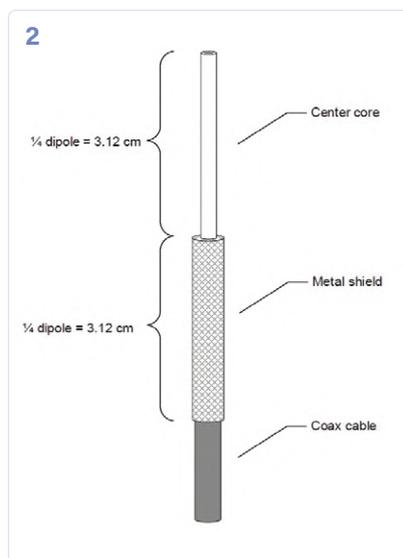
Dabei ist c die Lichtgeschwindigkeit (in m/s) und f die Frequenz (in Hz). Wie wir alle wissen, ist c gleich 299.792.458 m/s. Für 2,4 GHz WLAN ergibt dies eine Wellenlänge von 0,1249 m oder 12,49 cm. Dieser Wert ist wichtig, da die Antennen in Längen gebaut werden, die bestimmte Bruchteile der Wellenlänge ausmachen, zum Beispiel 1/4 der Wellenlänge (3,12 cm) oder die Hälfte (6,25 cm). Mit diesen Zahlen können wir das Aliexpress-Kabel so schneiden und nachbearbeiten, wie man es in **Bild 2** sieht.

Zuerst werden wie in **Bild 3** möglichst genau 3,12 cm der Kunststoffummantelung entfernt, so dass die Metallabschirmung (Geflecht) freigelegt ist. Das Geflecht wird nun ganz vorsichtig vom inneren Kunststoff des Kabels entflochten, leicht verdrillt und wie in **Bild 4** „rückwärts“ um den äußeren Kunststoffmantel gelegt. Damit haben Sie schon einmal glücklich eine einfache „Lambda/4-Stab- oder Monopolantenne“ gebaut.

Beim Experimentieren kann ein Hilfsprogramm wie WifiScan zeigen, ob es schon irgendwelche Verbesserungen gegenüber der internen Antenne eines WLAN-Routers gibt.

Am einfachsten ist es, die Antenne an einem festen Ort zu platzieren und dann zu sehen, wie viele WLAN-Access-Points zu sehen sind. Machen Sie dasselbe zum Beispiel mit einem Wemos-Board mit interner Antenne.

Nach mehreren Versuchen mit dem WifiScan-Programm und sorgfältigen Anpassungen der Metallab-



schirmung konnte ich eine Verbesserung der RSSI (Received Signal Strength Indication) feststellen. RSSI ist ein Begriff, der zur Messung der relativen Qualität eines empfangenen Signals zu einem Client-Gerät (in meinem Fall einem ESP8266) verwendet wird, aber keinen absoluten Wert hat. Der RSSI-Wert variiert auch stark zwischen den Chipsatzherstellern und kann daher nur anzeigen, ob die Experimente mit der Antenne in die richtige Richtung gehen. ◀

200207-02



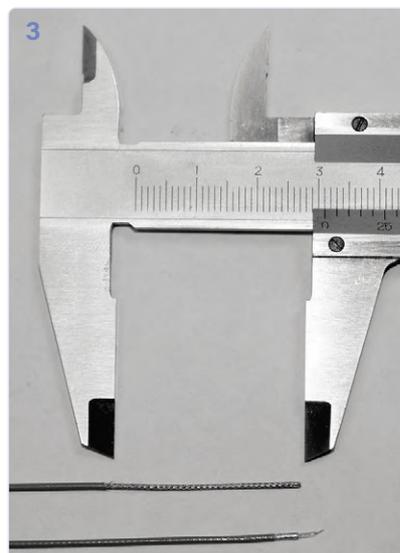
IM ELEKTOR-SHOP

> Buch „IoT Home Hacks with ESP8266“

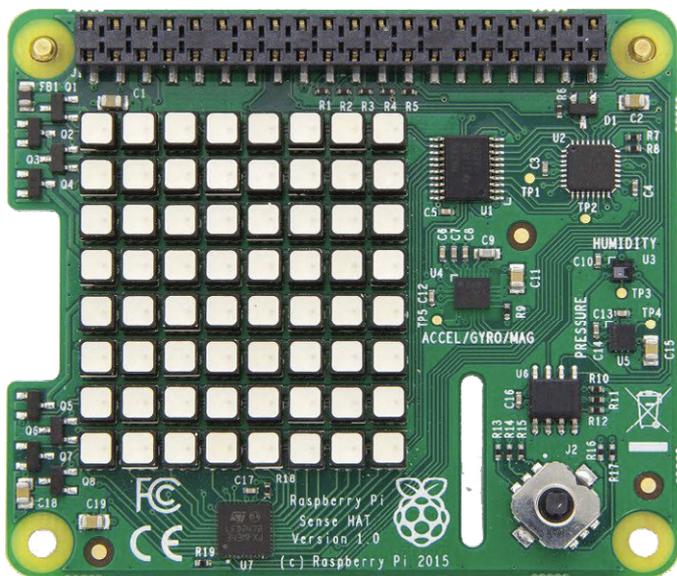
www.elektor.de/iot-home-hacks-with-esp8266

> E-Book „IoT Home Hacks with ESP8266“

www.elektor.de/iot-home-hacks-with-esp8266-e-book



Einfacher Ein-Aus-Temperaturregler mit Raspberry-Pi-HAT



Von **Dogan Ibrahim**

Die Sense-HAT-Platine wird normalerweise an den 40-poligen GPIO-Verbinder des Raspberry Pi angeschlossen. Um neben der Sense-HAT-Platine auch weitere Komponenten mit dem Raspberry Pi zu verbinden, muss der Sense-HAT mit dem Raspberry Pi via Flachbandkabel oder mit Jumperkabeln verbunden werden, damit die anderen Anschlüsse des Raspberry Pi zugänglich sind. Daher ist es wichtig zu wissen, welche Pins der Sense-HAT-Platine mit dem Raspberry Pi verbunden werden müssen und welche Pins des Raspberry Pi frei sind. Das Sense-HAT-Board besteht aus sieben „Hauptbauteilen“ und einer LED-Matrix. Diese Bauteile sind:

Bauteil	I ² C-Busadresse	Funktion
HTS221	0x5F	Feuchtigkeitssensor
LPS254H	0x5C	Druck/Temperatursensor
LSM9DS1	0x1C, 0x6A	Beschleunigungsmesser + Magnetometer
SKRHABE010	-	Joystick
LED2472G	0x46	LED-Matrix-Controller
LED-Matrix	-	-
ATTINY88	-	Atmel-Mikrocontroller

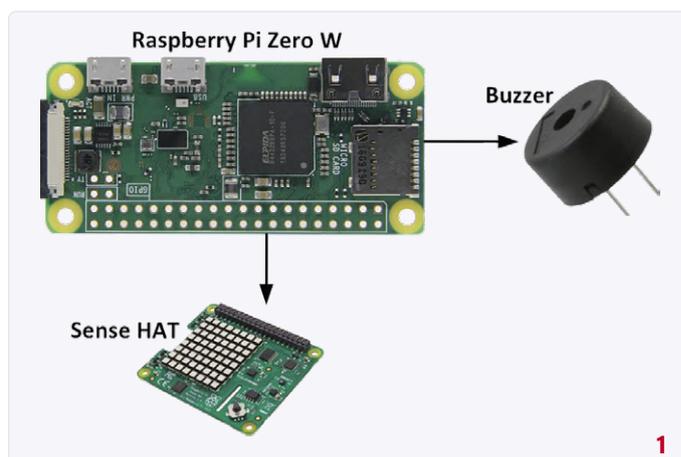
Diese Bauteile werden über den I²C-Bus unter den aufgeführten Adressen gesteuert. Zusätzlich zu den I²C-Steuerleitungen kann der ATtiny88-Mikrocontroller über die auf dem Board befindlichen SPI-Bus-Steuerleitungen (MOSI, MISO, SCK, CE0) programmiert werden. Rechnet man die Stromversorgung hinzu, sind also für den Anschluss der Sense-HAT-Platine an den Raspberry Pi nur neun Pins des GPIO erforderlich:

Sense-HAT-Pin	Raspberry-Pi-Pin	Funktion
3	3 (GPIO2)	SDA (I ² C)
5	5 (GPIO3)	SCL (I ² C)
1	1 (+3,3V)	3,3-V-Versorgung
9	9 (GND)	Versorgungsmasse
2	2 (+5V)	5-V-Versorgung
16	16 (GPIO23)	Joystick
18	18 (GPIO24)	Joystick
27	27 (ID_SD)	EEPROM
28	28 (ID_SC)	EEPROM

Wie Sie leicht herausfinden können, belässt der Sense-HAT eine große Anzahl von Raspberry-Pi-GPIO-Pins zur freien Verwendung. Unser kleines „Projekt“ ist ein On-Off-Temperaturregler, bei dem der Sense-HAT mit einem Raspberry Pi Zero W verbunden ist, um die Umgebungstemperatur zu messen. Zusätzlich ist ein kleiner Summer an einem Raspberry Pi-Port angeschlossen. Der Temperaturwert ist im Programm fest eingestellt (hardcodiert). Wenn die Umgebungstemperatur niedriger als die eingestellte Temperatur ist, wird der Summer aktiviert und die LED-Matrix zeigt die Umgebungstemperatur in rot an. Ist die Umgebungstemperatur hingegen höher als der eingestellte Temperaturwert, so wird der Summer deaktiviert und die Temperatur blau angezeigt. Der Summer kann in diesem Projekt leicht durch ein Relais ersetzt werden, das der Steuerung einer Heizung oder eines Lüfters dient. Die Heizung schaltet sich dann ein, wenn die Umgebungstemperatur niedriger als der eingestellte Wert ist, der Lüfter verhält sich genau umgekehrt.

Bild 1 zeigt das die Beteiligten dieses Projekts und **Bild 2**, wie sie miteinander verbunden werden. Der Summer ist an den Portpin GPIO4 des Raspberry Pi angeschlossen, und zwar genau wie die Sense-HAT-Platine über Jumperdrähte.

Der Code für das Steuerprogramm *tempcont.py* ist in **Listing 1** zu sehen. Es kann kostenlos von [1] heruntergeladen werden. Zu Beginn des



Wie macht man (schöne) Fotos von Platinen ...

... und elektronischen Bauteilen?



Von **Mariline Thiebaut-Brodier, Jean-Paul Brodier**
und der Grafikabteilung von Elektor
Bildmaterial: **Scheimpflug**

Beim Erstellen eines Elektor-Artikels ist es für unsere Autoren nützlich, wenn sie auch gut mit einer Kamera umgehen können, um die von ihnen beschriebenen Platinen mit einem professionell anmutenden Foto zu dokumentieren. Hier ein paar Tipps.

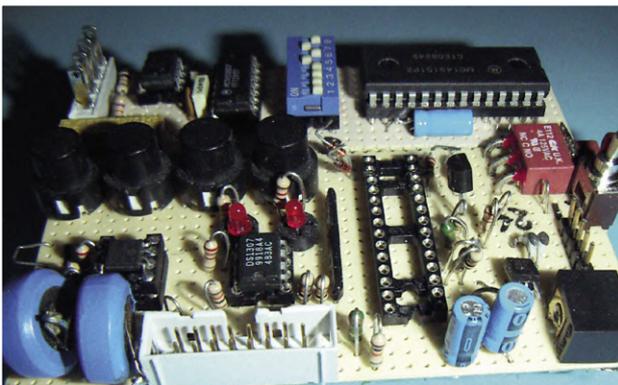


Bild 1. Aufgenommen ohne Beachtung unserer Tipps.

Ein gutes Foto eines elektronischen Bauteils oder einer Platine muss kontrastreich sein, und die Beleuchtung sollte die Konturen ohne dunkle Schatten hervortreten lassen. **Bild 1** zeigt das Ergebnis, das man mit einer Kamera für Amateure mit eingebautem Blitz erhält.

Die Fehler springen förmlich ins Auge: Die Helligkeit im Hintergrund ist unzureichend, die Schatten stören und die Einzelheiten sind nur undeutlich zu erkennen. Außerdem erscheint die vordere Kante der Platine gekrümmt. Die folgenden Abschnitte zeigen Ihnen, wie man es besser machen und ein Ergebnis wie in **Bild 2** erzielen kann.

Beleuchtung

Die auffälligsten Beleuchtungsfehler sind oft am leichtesten zu beheben. Verwenden Sie niemals das in der Kamera eingebaute Blitzlicht, da dies immer zu harten, dunklen Schatten führt. Eine wesentlich gleichmäßigere Schattierung erhält man durch Verwendung mehrerer, nicht punktförmiger, weicherer und diffuser Lichtquellen wie zum Beispiel Schreibtischlampen und kleineren Spots.

Das Foto aus **Bild 3** zeigt eine einfache Lösung für ein gleichmäßig beleuchtetes Bild: Eine Standlupe fürs Elektroniklabor mit ringförmiger Neonröhre. Die Linse wurde entfernt, damit man durch den Ring hindurch fotografieren kann. Mit Linse wäre das Bild stark verzerrt.

Um die scharfen Schatten um den Rand der Platine zu vermeiden, empfiehlt es sich, diese mit einem kleinen Gegenstand (wie zum Beispiel einem Radiergummi oder mit leicht anzupassender Knetmasse) von der Unterlage (hier ein grünes Blatt Papier) ein wenig abzuheben. Um die Lichtstreuung zu optimieren, fügen wir noch einen aus zwei zusammengehefteten, weißen DIN A 3 oder DIN A 4 Bögen bestehenden Zylinder um den Ring der Lampe hinzu (**Bild 4**).

Da das Licht der verwendeten Neonröhre schwächer als das eines Blitzes ist, ist in der Regel eine längere Belichtungszeit notwendig. Daher sollte die Kamera zur Vermeidung verwackelter Aufnahmen

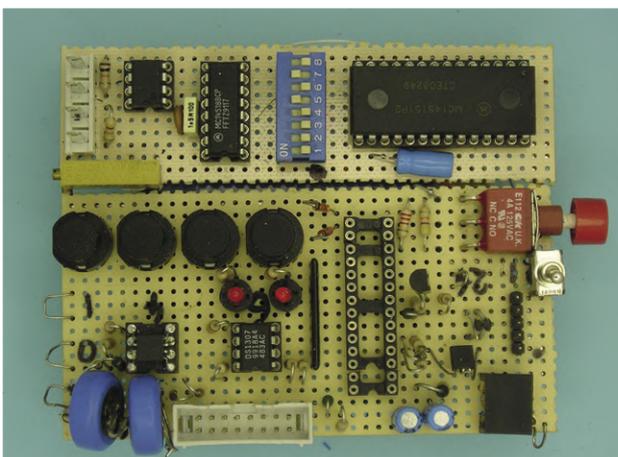


Bild 2. Aufgenommen mit Beachtung unserer Tipps ;-).

unbedingt auf einem Stativ montiert und damit fixiert werden. Ein Stativ sorgt außerdem dafür, dass auch alle weiteren Aufnahmen den gleichen, zuvor optimal gewählten Bildausschnitt besitzen. Wenn die Kamera an einem Galgen mit Gelenk befestigt ist (**Bild 5**), lässt sie sich leicht in die erforderliche, senkrecht nach unten gerichtete Position bringen. Wer nicht über die entsprechende Ausrüstung verfügt, kann sich auch, wie in Bild 3 gezeigt, mit gelochten Winkelschienen aus dem Baumarkt behelfen, wobei sich aus diesen, je nach Geschick und Vorlieben, auch die unterschiedlichsten, fest montierten, verschraubten oder verschweißten Konstruktionen zur dauerhaften Anwendung herstellen lassen. Der schwarze Bohrmaschinenkoffer und die Schraubzwinge dienen dabei natürlich nur als provisorische Hilfsmittel für einen ersten Versuch – genau so wie die zur Justierung unter den Koffer gelegte Platte. Achten Sie darauf, dass die Unterseite der Kamera durch die Montage an der Schiene nicht zerkratzt wird (**Bild 6**). Der Kamerafuß besitzt ein Gewinde von $\frac{1}{4}$ Zoll mit einer Steigung von 20 Gängen pro Zoll (UNC). Statt einer herkömmlichen, auf der Befestigungsschraube sitzenden Gewindemutter empfiehlt sich die Verwendung einer Flügelmutter. Sollten Sie kein passendes Exemplar im Baumarkt finden und sich im Umgang mit einem Schweißbrenner auskennen, so können Sie sich das benötigte Exemplar auch selber herstellen (**Bild 7**): Sägen Sie dazu zwei um 180 Grad voneinander entfernte Schlitzlöcher in eine passende Mutter, stecken Sie zwei der vier Teile einer mit der Metallsäge geviertelten Unterlegscheibe in die betreffenden Schlitzlöcher und löten Sie diese mit einem Gasbrenner fest. Vergessen Sie nicht, zum Abschluss noch einen Weißabgleich der Kamera bei künstlicher Beleuchtung durchzuführen, da die Fotos sonst zu gelbstichig werden könnten.

Verzerrungen und Blenden-Einstellung

Auf dem ersten Bild (Negativbeispiel) erscheint die Platine nach oben gewölbt. Solche geometrischen Verzerrungen lassen sich durch Veränderungen der Brennweite in gewissen Grenzen verhindern. Mit der in der Kamera enthaltenen Zoom-Funktion sollte die Brennweite auf ihren längsten Wert eingestellt werden (Tele). Zwar kommt es zu einer leichten Kissenverzeichnung, doch diese fällt nicht so markant aus



Bild 4. Mit ein paar Blatt Papier kann die Verteilung des Lichtes deutlich verbessert werden.



Bild 5. Stativ mit Galgen in Aktion.

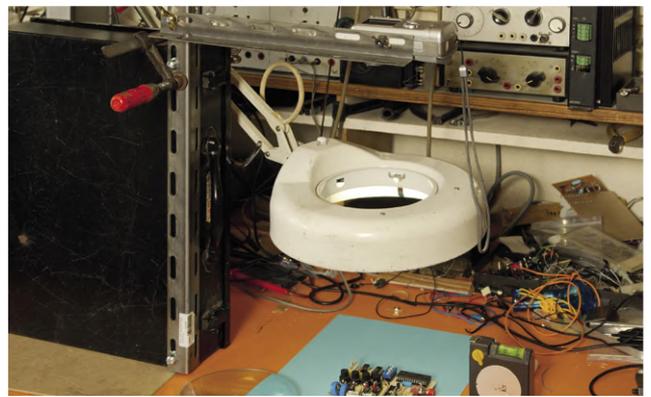


Bild 3. Die Lupe dieser Lampe wurde demontiert, um mit der Kamera durch den Ring hindurch fotografieren zu können.

wie die tonnenförmige Verzeichnung bei einer kurzen Brennweite (Weitwinkelfunktion). Wir haben bei Objektiven mit variabler Brennweite leider nur die Wahl zwischen Pest und Cholera.

Eine möglichst lange Brennweite verleiht dem Betrachter das Gefühl, dem Objekt nahe zu sein, in das Objekt einzutauchen und es berühren zu wollen. Das handelsübliche Stativ bietet hier den Vorteil, dass man in der Einstellung „Tele“ die notwendigen Nachregulierungen des Abstandes zwischen Kamera und Objekt leicht durchführen kann.

Oft lässt auch die Bildschärfe zu wünschen übrig. Es ist durchaus möglich, alle Objekte vom Vordergrund bis zum Hintergrund der zu fotografierenden Szene scharf abzubilden, doch dazu sollte vor allem bei professionellen Aufnahmen mit manuell einstellbarer Blende gearbeitet werden. Für ein vollständig scharfes Bild muss die Blende auf Werte zwischen 5 und 8 eingestellt sein.

Soll nur eine einzelne Bild-Ebene bzw. ein Detail scharf dargestellt werden und



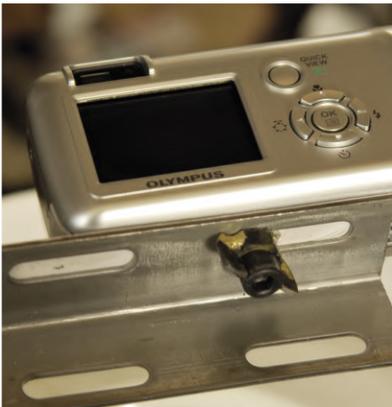


Bild 6. Schützen Sie die Kamera mit Klebeband vor Kratzern.



Bild 7. Selbstgemachte Flügelschraube.



Bild 8. Damit die Schraube zur Stabilisierung des Galgens bei jeder Neujustierung nicht erneut extrem fest angezogen werden muss, wurde mit einer Metallsäge ein Blechstreifen ausgeschnitten und dieser nach außen gebogen.

der Rest verschwommen bleiben, so muss die Blende weiter geöffnet und auf einen Wert von zum Beispiel 2,8 eingestellt sein.

Bildkomposition

Überlegen Sie sich vor einer Aufnahme, ob Sie das ganze Objekt oder nur einen bestimmten, wichtigen Teil davon aufnehmen. Bei Fotos, die vorwiegend auf einem Computerbildschirm betrachtet werden, wählen Sie am besten das Querformat. Lassen Sie oben und unten sowie an beiden Seiten des zu fotografierenden Objektes noch genügend Hintergrundfläche erscheinen. Wenn es um eine rein technische Illustration geht, sollte das Objekt ohne Perspektive aufgenommen werden, also von oben in flacher Darstellung (Bild 2). In allen anderen Fällen kann eine dreidimensionale Darstellung ein Objekt recht gut zur Geltung bringen, wobei der Betrachtungswinkel durch Ausprobieren so gewählt werden sollte, dass der zu fotografierende Gegenstand (Platine) am vorteilhaftesten erscheint (Bild 9). Um einen Eindruck der Größenverhältnisse zu vermitteln, kann es gelegentlich nützlich sein, dem Bild ein kleines Referenzobjekt wie zum Beispiel ein Streichholz oder Geldstück hinzuzufügen (Bild 10).

AUFLÖSUNG UND BILDFORMAT

Wir werden oft gefragt, wie hoch die Auflösung eines Bildes sein muss, damit es in guter Qualität gedruckt werden kann. Die Größe eines abdruckenden Bildes hängt nicht zuletzt von der Qualität der verwendeten Kamera ab: Je größer die Abbildung, desto mehr Megapixel sind notwendig. Die Tabelle unten zeigt die Relation zwischen der Zahl der Megapixel einer Kamera und der maximalen Darstellungsgröße, bezogen auf eine Qualität von 300 dpi (dots per Inch), ein für Magazine wie Elektor üblicher Wert.

Auflösung der Kamera (Pixel)	Maximale Abbildungsgröße bei 300 dpi
640 × 480	5,42 cm × 4,06 cm
800 × 600	6,77 cm × 5,08 cm
1024 × 768	8,67 cm × 6,5 cm
1280 × 960 (1,3 Megapixel)	10,84 cm × 8,13 cm
1600 × 1200 (2,1 Megapixel)	13,55 cm × 10,16 cm
1800 × 1200 (2,3 Megapixel)	15,24 cm × 10,16 cm
2048 × 1536 (3 Megapixel)	17,34 cm × 13 cm
2400 × 1600 (4 Megapixel)	20,32 cm × 13,55 cm

SCREENSHOTS

Für Screenshots empfehlen wir folgende Vorgehensweise:

- › Machen Sie das betreffende Fenster so groß wie möglich.
- › Machen Sie einen Screenshot (mit der „Druck“-Taste für den ganzen Bildschirm und mit der Druck- + Alt-Taste für das aktive Fenster).
- › Öffnen Sie das Bild (meistens mit Strg+V) in einem Bildbearbeitungsprogramm (z.B. Paint).
- › Speichern Sie das Bild im Format .BMP, .PNG oder .TIF und geben sie

ihm einen eindeutig identifizierbaren Namen.

- › Verwenden Sie nicht das JPG-Format (JPEG), da es hier zu viele Kompressionsartefakte geben kann.
- › Ändern Sie nicht die Farben.
- › Ändern Sie nicht das Seitenverhältnis.
- › Ändern Sie nicht die Auflösung.
- › Beschneiden Sie das Original nicht.

Mit anderen Worten: Ändern Sie nichts am Bild und schicken Sie es uns im Originalzustand.

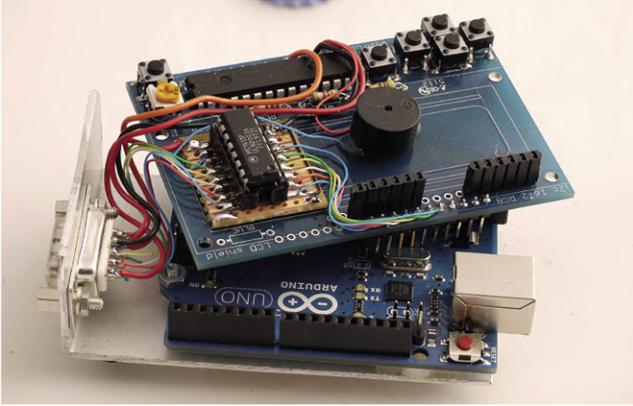


Bild 9. Neben der flachen Ansicht einer Platine vermitteln auch dreidimensionale Darstellungen gestapelter Platinen einen realistischen Eindruck.



Bild 10. Größenvergleich zweier Transistoren.
Tipp: Zur Vermeidung von Reflexionen auf Metalloberflächen sollte man diese vor der Aufnahme leicht mit Fett bestreichen.

Hintergrund

Fotografieren Sie das Objekt stets vor einem neutralen Hintergrund mit einer nicht zu aufdringlichen Farbe, die im Kontrast zum fotografierten Objekt steht. Der Hintergrund (Blatt Papier) sollte homogen sein, das heißt ohne verwirrende Linien, Karos oder Muster, die vom Objekt ablenken. Fotografieren Sie Ihre Platinen auch niemals auf einer Tischdecke oder einem gemusterten Bodenbelag, um nur zwei „verbotene“ Beispiele zu nennen. Bei automatischen Aufnahmen sollte kein weißer Hintergrund verwendet werden, da die Automatik dadurch verwirrt werden und das Ganze für eine zu kompensierende Gegenlicht-Aufnahme halten könnte. Das einzig Interessante und Sehenswerte auf dem Foto ist das Objekt selber. Aus diesem Grunde sollen auch keine Gegenstände oder Möbel aus der Umgebung auf dem Foto sichtbar sein.

Zusammenfassung

Dieser Artikel hat Ihnen vielleicht Lust gemacht, sich weiter in das Thema einzuarbeiten. Mit etwas Phantasie und zusätzlichem Material aus Ihrer Bastelkiste dürfte es nicht schwer fallen, die Qualität Ihrer Aufnahmen noch weiter zu verbessern.

Zögern Sie nicht, Ihre Tipps und Tricks zum Thema „Fotografieren von Elektronik“ mit anderen Elektor-Lesern zu teilen und senden Sie uns Ihre schönsten Fotos zu (redaktion@elektor.de)! ◀

(160463)

ÜBERSENDUNG DER DATEIEN

Einige Autoren senden uns Word- oder Open-Office-Dateien, in denen bereits alle Illustrationen im betreffenden Text an den richtigen Stellen eingebaut sind. Solch ein Dokument vermittelt zwar einen guten Eindruck, wie das Ganze später aussehen soll, aber wir benötigen Texte und Bilder bzw. Anhänge (Fotos, PDFs, Software, Zeichnungen) in getrennter Form. Am besten packen Sie alle Bild- und Textdateien in einen Zip-Ordner. Dabei ist es wichtig, dass alle zum Projekt gehörenden Dateien wie zum Beispiel Bilder eindeutig im Text referenziert sind, beispielsweise anhand einer Nummer. Wenn der Artikel bereits eine sechsstellige Artikelnummer besitzt, sollten auch die Filenamen aller zugehörigen Dokumente diese Nummer enthalten, zusammen mit einer Information zum Datei-

Inhalt. Vergessen Sie dabei auch bitte nicht die Versionsnummer. Wenn die Artikelnummer noch nicht existiert, kann das Projekt durch eine Abkürzung gekennzeichnet werden, ergänzt durch Ihre Initialen.

Bitte bearbeiten Sie die Fotos nicht selber, sondern schicken Sie sie uns so, wie sie in Ihrer Kamera vorliegen: unretuschiert. Die Bilder sollten auch nicht, wenn sie nicht allzu groß sind, komprimiert werden – zum Beispiel ins jpg-Format, da hierdurch stets gewisse Qualitätsverluste zu befürchten sind.

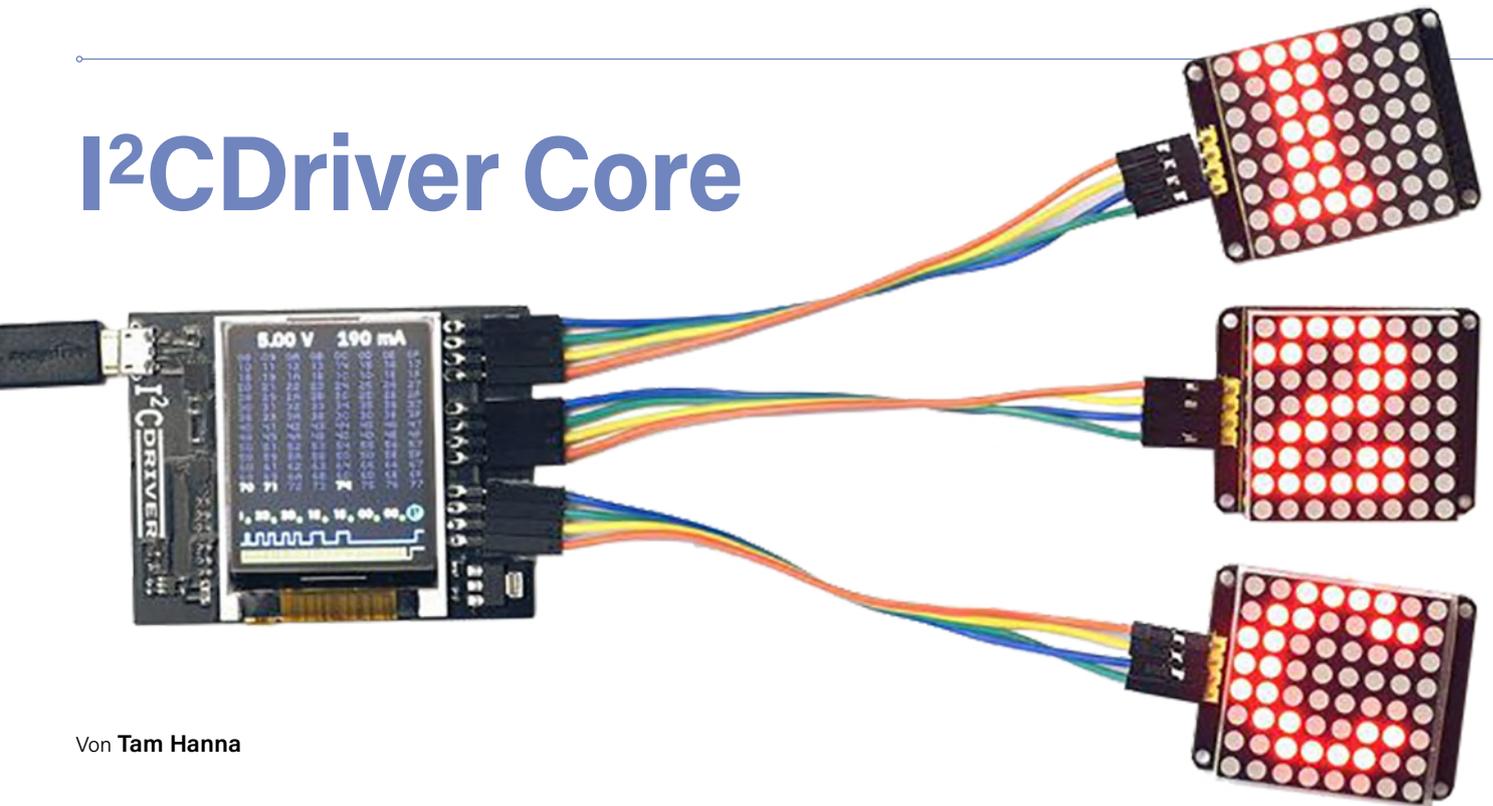
Umfangreiche Bilddateien können Sie über unseren FTP-Server hochladen. Nehmen Sie dazu Kontakt mit unserer Redaktion auf.

WEBLINKS

Leser, die des Französischen mächtig sind, finden hier weitere, wertvolle Hinweise:

- [1] Herstellung der Fotos für das Buch „Réalisez et programmez 12 applications pratiques pour maîtriser le PICBASIC PB-3B“ <http://thumblng.free.fr/MakingOuf/thumb.html>
- [2] Foto von Transistoren: <http://thumblng.free.fr/MakingOufTransi/thumb.html>

I²C Driver Core



Von **Tam Hanna**

I²C ist ein weit verbreiteter Bus, der in den unterschiedlichsten Embedded-Anwendungen zum Einsatz kommt. Für Test- und Entwicklungszwecke hat Excamera Labs mit dem I²C Driver ein Board entwickelt, das als Brücke zwischen PC-Software und I²C-Hardware (wie zum Beispiel Sensoren) dient. Die I²C-Kommunikation wird auch mitgeschnitten und auf einem Farb-Display übersichtlich dargestellt.

Nach dem Auspacken sieht man – wie in **Bild 1** gezeigt – eine Platine mit einem MicroUSB-Anschluss und drei gleichberechtigten Pinheadern mit den I²C-Signalen. Auf der Unterseite der Platine befindliche Gummipoppen verhindern ein Verrutschen auf dem Tisch.

Für das I²C Driver Core Kit legt Excamera Labs drei Dupontkabelblöcke bei, die die direkte Verbindung mit Hardware ermöglichen. Die Versorgungspins liefern nominell bis zu 500 mA bei 3,3 V – in **Bild 2** bis **Bild 4** ist zu sehen, wie sich das Produkt bei Kontakt mit einer elektrischen Last verhält.

Die 3,3 V entstehen übrigens aus der Stromversorgung des Rechners – es ist gut möglich, dass ein Teil der in den Oszillogrammen sichtbaren Welligkeit auf den USB-Bus zurückzuführen ist. Beachten Sie in diesem Zusammenhang auch, dass der I²C Driver keine galvanische Isolation zwischen DUT und Rechner garantiert.

Probelauf mit echter Hardware

Da im Unternehmen des Autors mit dem HygroSage gerade ein I²C-Sensorsystem entwickelt wird, bietet sich dieses für einen Test an. Hierzu schalten wir den I²C Driver zwischen PC und Sensorsystem (siehe **Bild 5**). Ob des geringen Stromverbrauchs „riskieren“ wir die Versorgung direkt aus dem I²C Driver.

HygroSage startete trotz des etwas heldenhaften Kabelaufbaus problemlos, die Stromverbrauchsanzeige auf der Oberseite des Bildschirms aktualisierte sich. Interessanterweise bevölkerte sich der Bildschirm des I²C Drivers erst nach dem Start der im nächsten Abschnitt genauer beschriebenen Computersoftware, die mitunter beim Hochfahren hängen blieb. Im Fall eines erfolgreichen Starts sieht man die in **Bild 6** gezeigte Heatmap, die über die Zugriffshäufigkeit auf verschiedene Geräte informiert.

Bei der Arbeit mit dem Farbdisplay fällt auf, dass Excamera den Einblickwinkel so gewählt hat, dass die Bildschirmausgabe aus einem gewissen Winkel sichtbar ist – blickt man direkt von vorne auf das Display, so ist insbesondere die Heatmap kaum zu sehen.



Bild 1. Der I²C Driver ist angenehm kompakt.

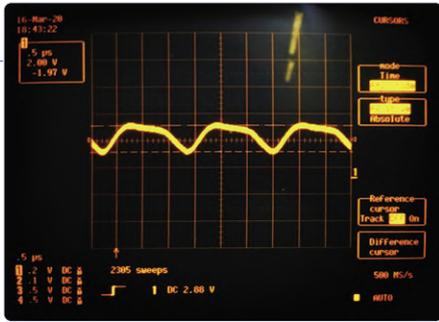


Bild 2. Der I²C Driver bei 500 mA...

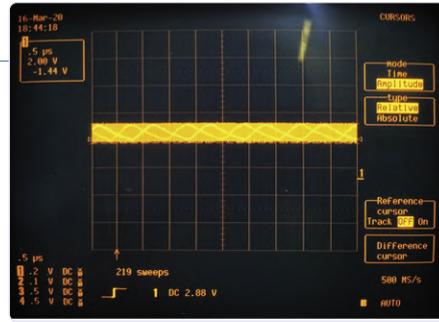


Bild 3. ... bei 200 mA ...

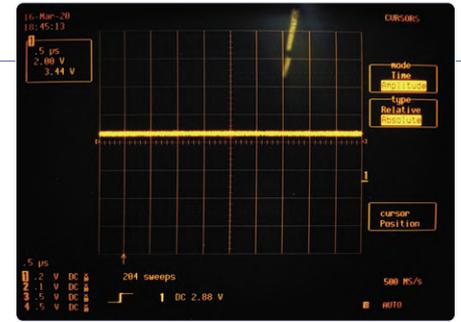


Bild 4. ... und im Leerlauf.

Steuere mich!

Unter [1] findet sich im Tab *Resources* die Datei *i2cdriver-installer.exe*, die die Inbetriebnahme unter Windows erlaubt. Nach dem Download müssen Sie sie rechts anklicken und im Einstellungsdialog als vom lokalen Rechner stammend markieren, bevor das Betriebssystem die Abarbeitung des Installers erlaubt. Mit Linux oder Mac OS arbeitende Entwickler finden auf der oben genannten Webseite analoge Anweisungen zur Einrichtung.

Nach getaner Arbeit öffnen wir den Ordner *C:\Program Files (x86)\Excamera Labs\I2C Driver*, in dem wir sowohl ein Kommandozeilenwerkzeug als auch eine GUI-Version des Produkts finden.

Wer die Software bei verbundenem I²C Driver anwirft und auf den Knopf *Monitor Mode* klickt, bekommt – wie in **Bild 7** gezeigt – Informationen über die letzte abgefeuerte Transaktion.

In der Praxis ist der Nutzen der (optisch ansprechenden) Analysefunktion allerdings eingeschränkt, weil in so gut wie allen Fällen mehr als nur ein Paket gleichzeitig durch die Luft fliegt. In diesem Fall bietet sich stattdessen ein Klick auf *Capture Mode* an. Der Knopf erscheint daraufhin, wie in **Bild 8** gezeigt, aktiviert.

Die – in Tests des Autors reproduzierbar abstürzende – Funktion ist insofern grenzgenial, als sie das aufwandsarme Mitschneiden von Kommunikation ermöglicht. „Haarige“ Fehler, die nur gelegentlich auftreten, lassen sich so analog zum DPO-Verfahren von Tektronix „totmessen“.

Interaktion, programmatisch

Vor einiger Zeit gab es bei der Entwickler-Schmiede des Autors einen Auftrag, der die Implementierung eines vergleichsweise komplexen Algorithmus erforderlich machte. Als angenehmster Weg erwies sich, das Verfahren im ersten Schritt am PC zum Laufen zu bekommen, um es danach auf den Controller zu transplantieren. Bei der Inbetriebnahme komplizierter Sensoren bietet sich eine ähnliche Vorgehensweise an. In der Eingabeaufforderung steht mit *i2ccl* ein Programm zur Verfügung, über das sie nach dem folgenden Schema Befehle an den I²C Driver senden:

```
C:\Program Files (x86)\Excamera Labs\I2C Driver>i2ccl.exe
Usage: i2ccl <PORTNAME> <commands>
```

Von besonderem Interesse ist hier die Möglichkeit, Informationen in einzelne Register von angeschlossenen Geräten zu schreiben oder diese auszulesen. Dies hilft nicht nur bei der Inbetriebnahme unbekannter Sensoren, sondern lässt sich auch zum Auslesen von Informationen bei (automatisierten) Testläufen verwenden. Wer nicht in der Shell programmieren möchte, kann auf eine Python-API setzen.

Der Hersteller demonstriert die Verwendung durch eine Gruppe vorgefertigter Beispieldriver – mit dem folgenden Snippet lässt sich beispielsweise ein LM75B abernten:

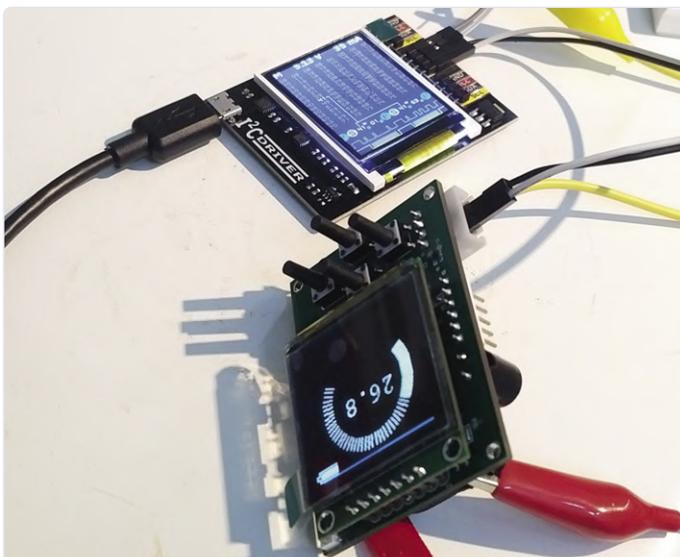


Bild 5. Der I²C Driver in Verbindung mit einem Sensorboard des Autors.

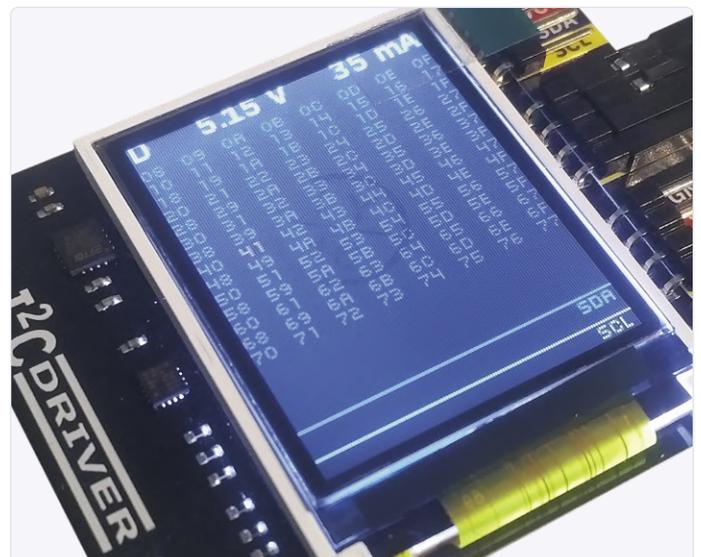


Bild 6. Wo nur ein Sensor, da nur eine farblich hervorgehobene Zahl.

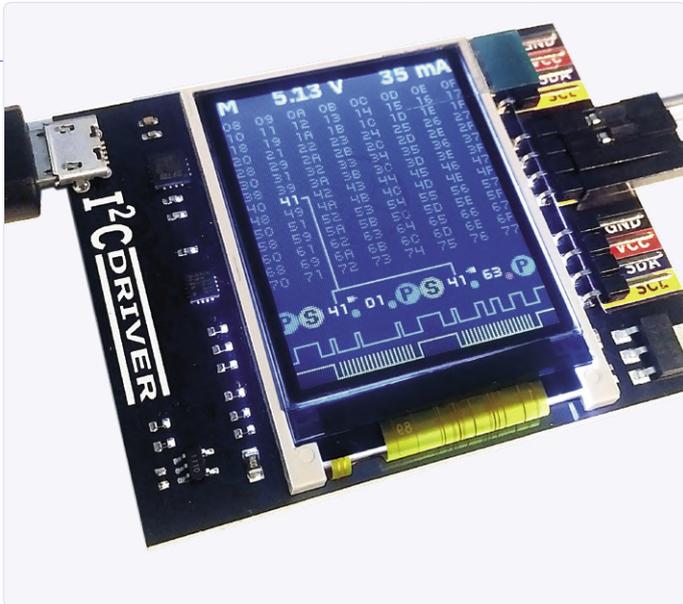


Bild 7. Registeroperationen scheinen am Display des I²C Driver auf.

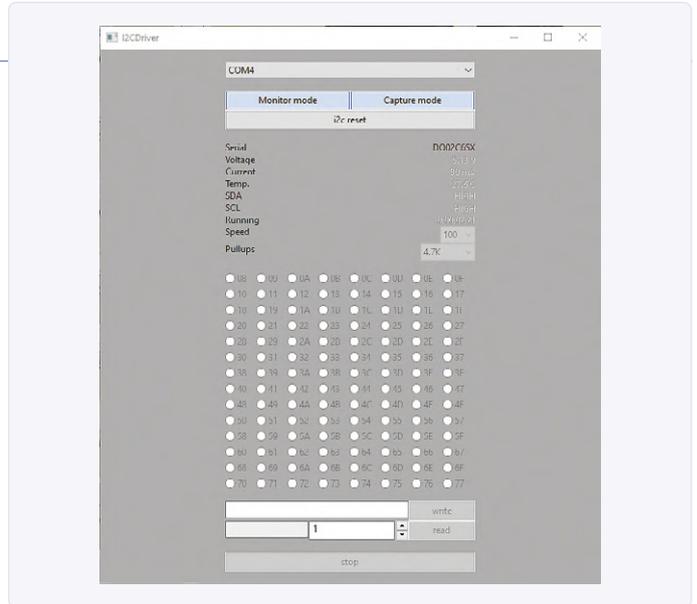


Bild 8. Die I²C-Driver-Desktopsoftware versprüht rustikalen Charme.

```
import i2cdriver
i2c = i2cdriver.I2CDriver("/dev/ttyUSB0")
d=i2cdriver.EDS.Temp(i2c)
d.read()
17.875
d.read()
18.0
```

Die eigentliche Steuerungs-API ist simpel, und steht auf GitHub [2] zur Einsichtnahme bereit:

```
class LM75B:
    def __init__(self, i2, a = 0x48):
        self.i2 = i2
        self.a = a
```

Excamera Labs implementiert die Hardwaretreiber unter Nutzung der Python-OOP-API. `self` ist ein von der Sprachspezifikation eingeforderter Treiber, während `i2` ein I²C-Treiberobjekt darstellt. Zu guter Letzt ist `a` die Adresse, unter der der Sensor anzusprechen ist. Das Einlesen von Registerinformationen erfolgt dann nach dem folgenden Schema:

```
def reg(self, r):
    return self.i2.regrd(self.a, r, ">h")

def read(self):
    return (self.reg(0) >> 5) * 0.125
```

Angemerkt sei, dass es ein an `i2cdetect` angelegtes Scan-Kommando gibt, das bei Aufruf aus der Python-Kommandozeile die von OrangePi und Co. bekannte Scanfunktion ausübt. Zu guter Letzt sei noch auf die unter [3] bereitstehende Dokumentation verwiesen. Sie erklärt sowohl die I²C-API als auch das physikalische Kommunikationsprotokoll – wer mit der FTDI-API auf Du und Du ist, kann den I²C Driver auch direkt ansprechen.

Fazit

Der I²C Driver ist eines jener Produkte, dessen Lebenssinn man erst nach etwas Kontemplation erfasst, dieses danach aber nicht mehr missen möchte. Egal ob bei der schnellen Analyse der Aktivität eines I²C-Netzwerks oder bei der Inbetriebnahme eines Sensors – das Board leistet wertvolle Hilfe. Der Preis ist in Anbetracht der einzusparenden Zeit angemessen – dass es keinen Standalone-Modus gibt, ist schade. ❏

200148-01



IM ELEKTOR-STORE

> I²C Driver Core: www.elektor.de/i2cdriver-core

WEBLINKS

- [1] Software: <https://i2cdriver.com/>
- [2] Steuerungs-API: <https://github.com/jamesbowman/i2cdriver/blob/master/python/lm75b.py>
- [3] Dokumentation: <https://i2cdriver.com/i2cdriver.pdf>

Touchscreen Joy-View 13

Von **Harry Baggen**

Es ist nicht so einfach, einen kleinen, aber qualitativ hochwertigen Monitor zu finden, der auch noch portabel ist. Das Modell Joy-View 13 erfüllt diese Kriterien. Dank seiner HDMI- und USB-C-Ports kann er an fast alle modernen Geräte angeschlossen werden. Wir haben dieses 13-Zoll-Display mit verschiedenen Computertypen getestet.

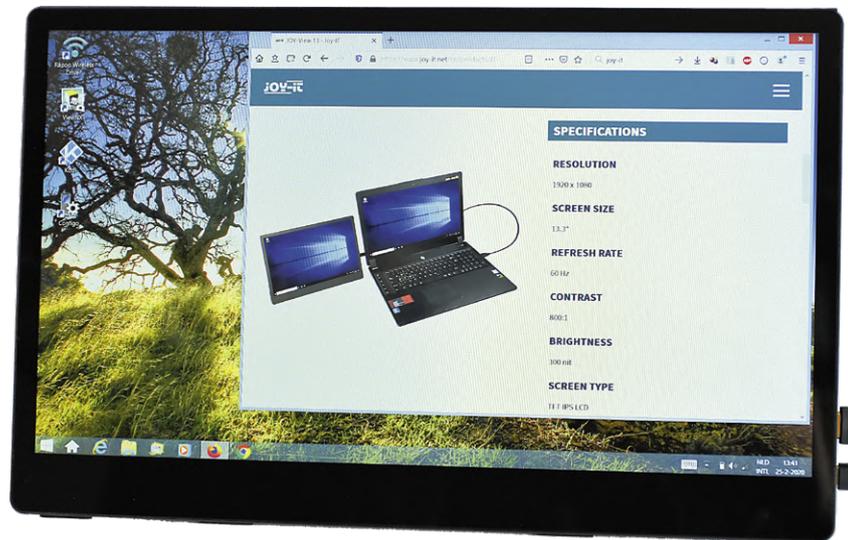


Bild 1. Der portable Touchscreen Joy-View 13.

Ein zweiter Bildschirm am Laptop kann in bestimmten Fällen sehr nützlich sein. Einfache Monitore gibt es viele, und sie sind nicht teuer, aber man kann sie nicht so leicht mitnehmen. Für solche Fälle empfiehlt sich ein Exemplar, das kompakt ist und „Touch“ hat sowie idealerweise auch noch unabhängig vom Stromnetz funktioniert. Genau das alles kann dieser kleine 13“-Bildschirm (Bild 1). Er ist dünn, robust, kann über das Stromnetz, den Laptop oder per Powerbank mit Strom versorgt und dank USB-C auch an modernste Laptops und Smartphones angeschlossen werden.

Auspacken

Nach dem Auspacken glaubt man zunächst, ein großes Tablet in den Händen zu halten. Das Gehäuse des Bildschirms besteht aus



Bild 2. Anschlüsse des Joy-View 13 rechts oben und links unten.

schwarz eloxiertem Aluminium und ist sauber verarbeitet. Der größte Teil des Bildschirms ist nur 5 mm dick. Das untere Drittel, wo sich die Anschlüsse und wahrscheinlich auch die Elektronik befinden, ist mit 9 mm etwas dicker (Bild 2). Der Bildschirm wird mit einer passenden Kunststoffabdeckung bzw. einen Smart Case geliefert, das magnetisch ist und daher auf beiden Seiten des Bildschirms ziemlich fest „klebt“ (Bild 3 und 4). Diese Abdeckung dient auch als Ständer, man kann sie (magnetisch) an der Rückseite befestigen und den unteren Teil nach hinten klappen (Bild 5). Um sicherzustellen, dass der Monitor nicht verrutscht, befinden sich auf der Unterseite vier kleine, rutschverhindernde Pins.

Der dickere untere Teil hat einen Ein/Aus-Schalter auf der linken Seite, einen Wippschalter für die Menübedienung und einen USB-C-Stromanschluss. Auf der rechten Seite befindet sich ein Mini-HDMI-Anschluss, ein USB-C-Anschluss für Strom und Daten sowie eine 3,5-mm-Klinkenbuchse für Kopfhörer. Im Sockel befinden sich auch kleine Löcher für die beiden eingebauten Lautsprecher. Einschließlich Abdeckung wiegt der Bildschirm knapp über 1 kg. Nicht wirklich leicht, aber dafür ziemlich solide.

Bei Joy-View 13 sind eine Reihe Kabel im Lieferumfang: Ein HDMI/Mini-HDMI-Kabel, ein USB-C/USB-C-Kabel, ein USB-A/USB-C-Kabel, ein HDMI/Mini-HDMI-Adapter. Außerdem sind ein Netzteil und sogar ein Reinigungstuch für das Display im Karton.

Screen

Das eigentliche Display hat eine Diagonale von 13,3“. Die Auflösung ist Full-HD, hat also wie viele moderne Laptops die üblichen 1920 x 1080 Pixel. Die Vorderseite reflektiert ziemlich stark, aber das ist bei einem Touchscreen so gut wie immer der Fall. Die Blickwinkelstabilität des IPS-Bildschirms ist in allen Richtungen gut.

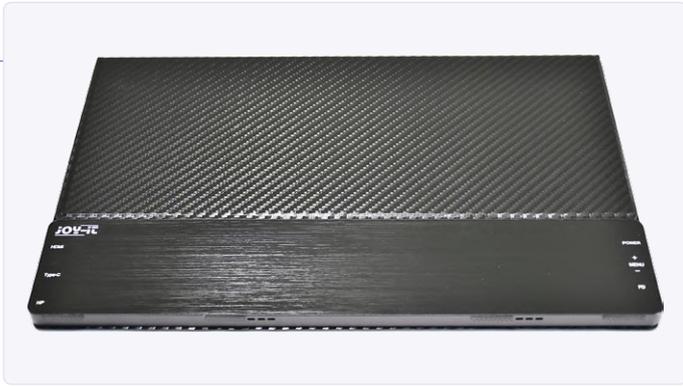


Bild 3. Die Rückseite des Gehäuses: Der geriffelte Teil ist das magnetische Smart Case.



Bild 4. Monitor mit komplett ausgeklapptem Smart Case.

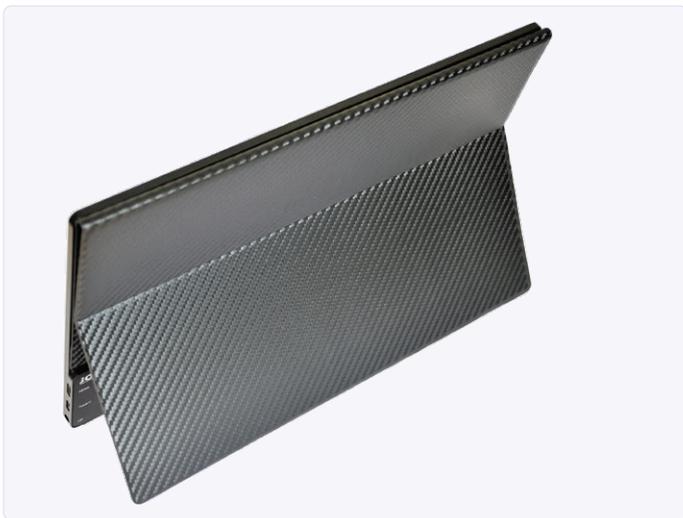


Bild 5. Das nach hinten ausgestellte Smart Case dient gleichzeitig als Ständer.

Aus großem Winkel wird das Bild zwar etwas dunkler, aber die Farben verändern sich kaum. Das Bild ist kontrastreich, und die maximale Helligkeit ist mit 300 Cd/m^2 recht gut – sie entspricht meinem Standardmonitor und dem eingebauten Bildschirm meines Laptops. Man kann damit sogar in einem hell erleuchteten Raum oder draußen im Schatten arbeiten. Lediglich in der prallen Sonne würde ich es nicht versuchen. Die Farbwiedergabe ist zwar gut, doch für mich sind die Farben subjektiv etwas zu gesättigt. Letzteres lässt sich über die Menüeinstellungen korrigieren. Das Menü bietet übrigens viele Möglichkeiten, aber die Bedienung ist nicht ganz trivial: Zur Aktivierung des Menüs muss man den Wippschalter betätigen; dazu muss man den vorstehenden Schalter recht tief drücken. Dies gilt auch für die Auswahl der Menüpunkte, weshalb ich manchmal den falschen Menüpunkt erwischte. Zum Glück braucht man das Menü nicht so oft. Wenn alles einmal eingestellt ist, benötigt man die Einstellungen nur noch selten.

Verbindungsoptionen

Dank zweier USB-C-Anschlüsse und eines Mini-HDMI-Ports kann der Monitor an viele Geräte vom PC bis zum Smartphone angeschlossen werden. Ich habe einige Kombinationen ausprobiert. Vor deren Beschreibung zunächst zur Stromversorgung dieses tragbaren Monitors: Dies kann über beide USB-C-Ports erfolgen, wobei der Anschluss auf der linken Seite nur für die Stromversorgung und der Anschluss auf der rechten Seite für Strom und auch für Daten ausgelegt ist. Außerdem kann das mitgelieferte Netzteil oder eine Powerbank verwendet werden. Zum Anschluss werden genügend Kabel mitgeliefert. Wenn der Monitor an einen USB-3.0-Port angeschlossen ist, kann dieser normalerweise sowohl die Stromversorgung als auch die Kommunikation mit dem Bildschirm übernehmen.

Noch ein Hinweis zur Touch-Fähigkeit. Dazu muss der Bildschirm über einen USB-C-Anschluss an ein modernes Smartphone, Tablet oder einen Computer angeschlossen werden (es gibt eine Liste kompatibler Geräte [1] auf der Webseite des Herstellers). Eine weitere Möglichkeit ist der Anschluss eines PCs, Laptops oder Raspberry Pi über zwei Kabel: Video per HDMI und Touchscreen-Steuerung via USB. Dies ist nicht nur mit Windows 8/10 möglich, sondern funktioniert auch unter Raspbian mit einem RPi. Zuerst habe ich den Monitor an meinen Desktop-PC unter Windows 10 angeschlossen. Dies funktionierte via HDMI-Kabel reibungslos. Die Stromversorgung erfolgte dabei durch das mitgelieferte Netzteil. Ich habe die Touch-Funktion nicht angeschlossen, da dies bei einer Dual-Monitor-Konfiguration mit aufgeteiltem Desktop und nur einem berührungsempfindlichen Bildschirm nicht sehr praktisch ist.

Als Nächstes habe ich den Joy-View 13 mit meinem Windows-8-Laptop ausprobiert. Auch das lief sehr gut. Nachdem ich das HDMI-Kabel eingesteckt und das USB-C-Kabel mit einem USB-3.0-Anschluss meines Laptops verbunden hatte, arbeitet auch die Touchscreen-Funktion sofort. Der Bildschirm reagierte direkt auf Berührungen, und Windows 8 ist – genau wie Windows 10 – gut auf Touchscreens vorbereitet.

Als letzte Kombination wurde der Monitor an einen Raspberry Pi 3B unter einer aktuellen Raspbian-Version angeschlossen (**Bild 6**). Auch hier erfolgte der Anschluss mit zwei Kabeln. Der Touchscreen funktionierte ebenfalls direkt, ohne dass ich etwas in Raspbian einstellen musste. Allerdings hatte ich Probleme mit



IM ELEKTOR-STORE

> JOY-iT Joy-View 13 Touchscreen
www.elektor.de/joy-it-joy-view-13-touchscreen

der Stromversorgung. Bei niedriger Auflösung konnte RPi gerade genug Strom via USB an den Bildschirm liefern, doch bei voller Auflösung ging das schief. Hier musste ich immer das Netzteil anschließen. Trotzdem bleibt: Es hat gut funktioniert! Ich habe auch versucht, den Bildschirm an ein ziemlich neues Tablet mit USB-C-Anschluss anzuschließen, doch hat das leider nicht geklappt. Dieses Tablet stand leider nicht in der Kompatibilitätsliste von Joy-iT.

Fazit

Der portable Touchscreen Joy-View 13 ist ein sauber verarbeiteter Bildschirm mit sehr guter Darstellung, eingebauter kapazitiver Touch-Fähigkeit und flexiblen Anschlüssen sowie Stromversorgungsoptionen. Dadurch kann er für viele Anwendungen und Computer-Kombinationen verwendet werden. Wenn Sie nach einem kompakten, stabilen und portablen Bildschirm suchen, ist er eine ausgezeichnete Wahl. Und wenn er ihnen zu klein sein sollte: Es gibt mit dem Joy-View 15 auch eine 15,6“-Version dieses Monitors. ◀

200169-01



Bild 6. Joy-View 13 an einem Raspberry Pi 3B unter Raspbian.

WEBLINKS

[1] Liste kompatibler Geräte: <http://joy-it.net/files/files/Produkte/JT-View13/JT-View13-Ger%C3%A4te.pdf>

Sie haben eine Idee für ein **Elektronikprojekt?
Oder schon **eine fertige Schaltung?**
Posten Sie Ihr Projekt unter
www.elektor-labs.de
(oder senden Sie es an redaktion@elektor.de).**

Auch kleine Projekte sind willkommen!
Elektor Labs: Von Lesern für Leser!

Elektor Labs: www.elektor-labs.de
Design, Build, Share... Electronics!

LED-Booster

für Mikrocontroller

Mit nur einem Bauteil!

Von **Martin Heine**

Immer wieder kommt der Wunsch auf, an einem mit 3 V betriebenen Mikrocontroller eine weiße LED anzuschließen. Doch der Versuch scheitert und die schöne weiße LED leuchtet nur ganz schwach oder überhaupt nicht...

Rote oder grüne LEDs jedoch funktionieren problemlos. Nach etwas Datenblatt-Recherche findet sich die Ursache prompt: Eine weiße LED hat eine typische Durchlass-Spannung von 3,2 V. Damit reichen 3 V einfach nicht aus, um ein helles Leuchten aus der

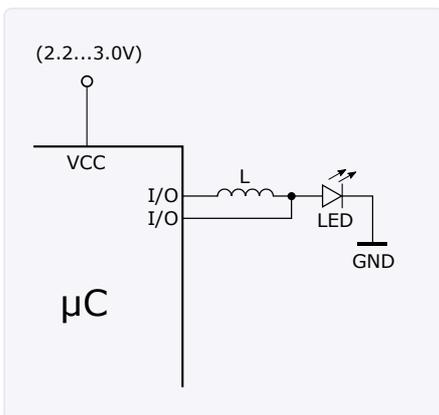


Bild 1. Schaltung des LED-Boosters.

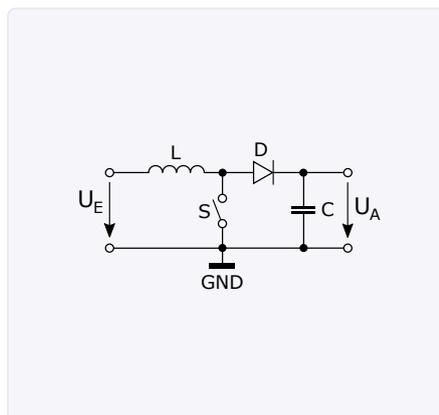


Bild 2. Prinzipschaltung des Boost-Konverters.

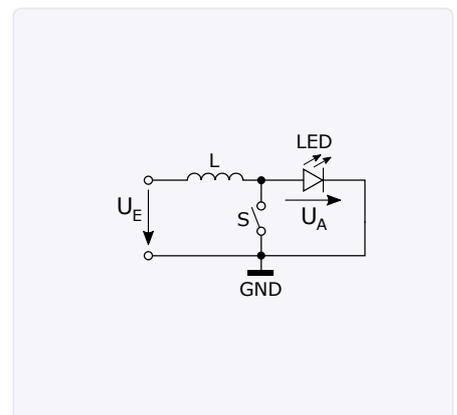


Bild 3. LED statt Diode und Kondensator.

LED zu zaubern. Man solle einen Boost-Konverter (Hochsetzsteller) zur Erzeugung einer höheren Spannung und einen Schalttransistor zum Schalten der LED einsetzen, so steht es oft in Internetforen. Doch für den Betrieb einer einzigen LED ist das wohl eher ein übertriebener Aufwand.

Die gute Nachricht: Es geht auch einfacher! Und zwar mit nur einem einzigen billigen zusätzlichen Bauteil: Einer Induktivität, die für ein paar wenige Cent erhältlich ist. Wird diese an der richtigen Stelle eingebaut und mit der richtigen Ansteuerung betrieben, leuchtet die weiße LED - und das sogar schon ab 2,5 V Versorgungsspannung des Mikrocontrollers. Magie? Natürlich nicht.

Bei der Schaltung aus **Bild 1** handelt es sich um einen Boost-Konverter. Manche sagen dazu im deutschen Sprachgebrauch auch „Aufwärtswandler“ oder „Hochsetzsteller“. Doch wie funktioniert dieser Minimal-Boost-Konverter?

Prinzip des Boost-Konverters

Eine Induktivität L (Spule), an deren Ende die Eingangsspannung U_E anliegt, wird mit einem Schalter S am anderen Ende gegen Masse geschaltet. Durch L fließt daraufhin ein Strom, der langsam ansteigt und dabei ein Magnetfeld erzeugt. Wird der Schalter S nach einiger Zeit geöffnet, bricht das Magnetfeld wieder zusammen, was in der Spule eine Induktionsspannung (**Bild 2**) entstehen lässt (ähnlich wie bei der Zündspule eines Fahrzeugs).

Diese Spannung addiert sich zur Versorgungsspannung. Somit liegt an der Diode D eine höhere Spannung U_A als die Versorgungsspannung an, mit welcher der Kondensator C geladen wird. Wird der Schalter S wieder geschlossen, wiederholt sich dieser Vorgang. Da eine LED im Prinzip nichts anderes ist als eine Diode, kann die Diode D auch durch eine LED ersetzt werden (**Bild 3**). Und diese kann dann direkt an GND angeschlossen werden.

Wie lässt sich nun dieser LED-Boost-Konverter an einem Mikrocontroller mit nur einem einzigen Bauteil, der Induktivität, realisieren? Dabei helfen die unterschiedlichen Port-Modi. Ein Mikrocontroller-I/O kann im Push-Pull- oder Open-Drain-Modus verwendet werden. Push-Pull bedeutet, dass bei einer logischen Eins (also Portpin auf High) der Pin gegen VCC geschaltet wird und bei einer logischen Null gegen GND (**Bild 4**). Bei der Open-Drain-Einstellung jedoch bleibt der

Ausgang bei einer logischen Eins offen und schaltet bei einer logischen Null nach wie vor auf GND, also genau so wie der Schalter S in unserem obigen Beispiel.

Die Größe der Induktivität hängt von der Schaltfrequenz, dem Strom, sowie der Ein- und Ausgangsspannung ab. In den meisten Datenblättern von Boost-Schaltreglern findet sich folgende Annäherung zur Bestimmung der Induktivität:

$$L = \frac{U_E \cdot (U_A - U_E)}{(\Delta I_L \cdot f_s \cdot U_A)}$$

- > L = Induktivität [H],
- > U_A = Ausgangsspannung [V],
- > U_E = Eingangsspannung [V],
- > I_A = Ausgangsstrom [A],
- > f_s = Schaltfrequenz [Hz],
- > ΔI_L = Induktor Ripple Current [A]

Der sogenannte *Induktor Ripple Current* ΔI_L gibt die Stromdifferenz zwischen minimalem und maximalem Spulenstrom an, also die „Peak-to-Peak Strom-Amplitude“ des Spulenstroms, der sich um den Ausgangsstrom bewegt (**Bild 5**).

Der Spulenstrom bewegt sich immer ein paar Prozent um den Ausgangsstrom herum. Auch dafür wird in den meisten Datenblättern von Boost-Schaltregler-Herstellern eine Annäherung angegeben, die für unseren LED-Booster völlig ausreichend ist:

$$\Delta I_L = 0,2 \cdot I_A(\max) \cdot \left(\frac{U_A}{U_E} \right)$$

Der maximale Ausgangsstrom $I_A(\max)$ ist der maximale Strom durch die LED, der erlaubt

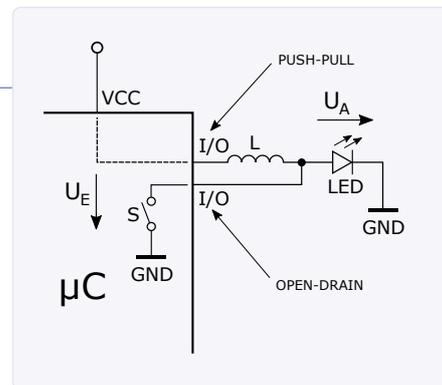


Bild 4. Parametrisierung der μC -Portpins.

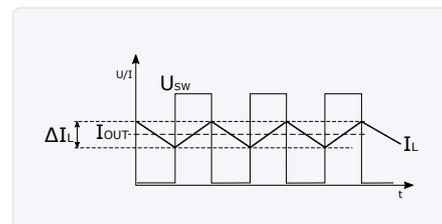


Bild 5. Verlauf von Schaltspannung und Strom.

wird. Die ausgewählte weiße LED ist mit 30 mA im Datenblatt angegeben. Wird für den Strom $I_A(\max)$ also 30 mA eingesetzt und für die Ausgangsspannung $U_A = 4$ V, sowie U_E (Eingangsspannung) = 2,5 V, dann kommt für $\Delta I_L = 9,6$ mA heraus. Damit errechnet sich nach oben angegebener Formel für eine Schaltfrequenz f_s von 1 MHz eine Induktivität L von 97,66 μH .

100- μH -Induktivitäten gibt es wie Sand am Meer. Sie sind in unterschiedlichen Bauformen erhältlich. Für eine kleine weiße LED mit einem typischen maximalen Strom von 30 mA eignet sich die SMD Bauform 0805.

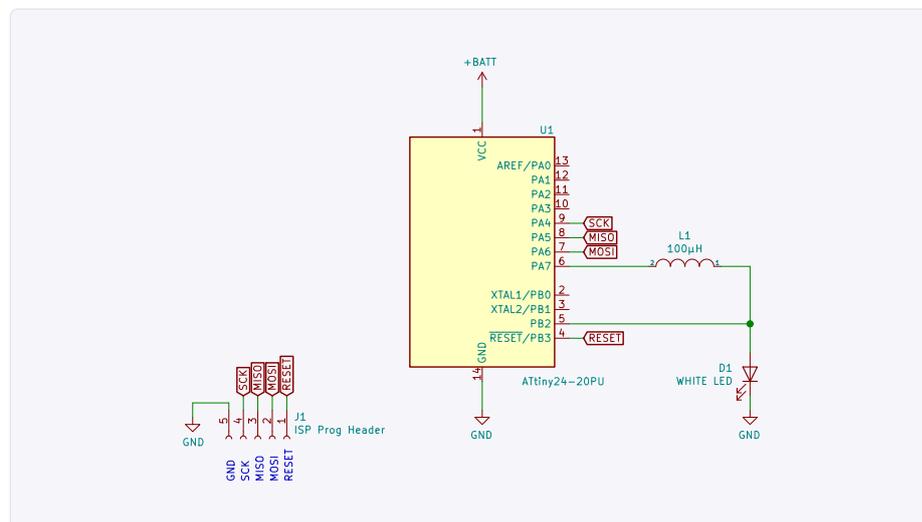


Bild 6. Schaltung des LED-Booster-Versuchsaufbaus.

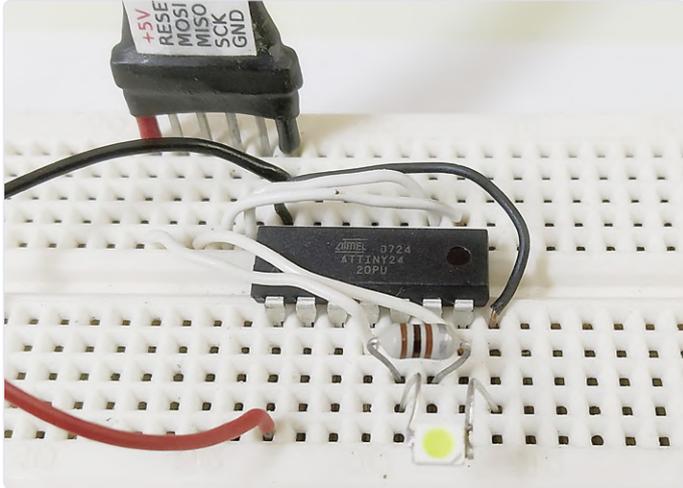


Bild 7. Versuchsaufbau des LED-Boosters.

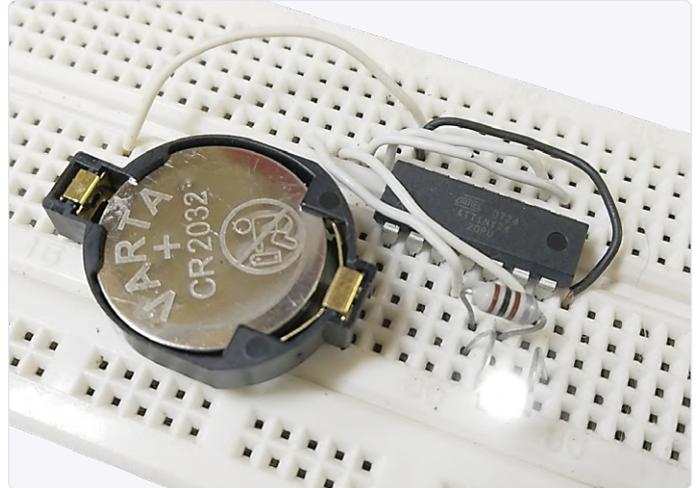


Bild 8. Weißer LED-Blinker mit 3-V-Knopfzelle.

Versuchsaufbau

Auf ein Steckbrett wird eine an zwei Drähte angelötete, weiße LED im PLCC-4-Gehäuse gesteckt, mit einer Durchlass-Spannung von 3,2 V und einem Durchlass-Strom von 30 mA. Dazu kommen ein AVR-Mikrocontroller ATtiny24 (der zufällig gerade in der Bauteilekiste lag) und eine 100-µH-Drossel (Bilder 6 und 7). Der Mikrocontroller wird an das Programmiergerät mit den Pins RESET, MOSI, MISO, SCK und GND angeschlossen. Ein einstellbares Labornetzteil versorgt die Schaltung mit 2 V bis 3 V.

Programmierung des LED-Boosters

Sind alle erforderlichen Verbindungen richtig angeschlossen, geht's ans Programmieren des ATtiny24. Rechteckspannungen können auf verschiedene Arten von einem Mikrocontroller ausgegeben werden. Zum Einen über PWM (Pulsweitenmodulation), zum Anderen über toggelnde I/O-Pins (Ausgänge), was der Einfachheit halber hier angewendet wird. Da für einen Boost-Konverter ein toggelnder Schalter gegen Masse benötigt wird, muss der entsprechende Ausgang also abwechselnd auf Open-Drain und Pull-Down gesetzt werden. Dies geschieht in einer Schleife direkt ohne Delays. Bei einem mit 8 MHz Takt betriebenen ATtiny ergibt das gerade so eine Ausgangsfrequenz von ungefähr 1 MHz.

Zum Ändern des voreingestellten internen 1-MHz-Takts auf 8 MHz muss die als Default gesetzte *CKDIV8*-Fuse ausgeschaltet werden. Beispielsweise mit dem Programm avrdude:

```
avrdude -U lfuse:w:0xEA:m
```

Wird eine zweite Schleife zum Toggeln des Versorgungs-Pins (U_E) um die Open-Drain-Pull-Down-Schleife (Booster-Schleife) gelegt, so entsteht ein klassischer Blinker mit weißer, „geboosteter“ LED (Bild 8). ◀

190361-01

Sourcecode des LED-Boosters

```
#define F_CPU 8000000UL // 8 Mhz

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    unsigned long int i;

    while (1)
    {
        PORTA |= (1<<PA7);
        DDRA |= (1<<DDA7); // LED ON
        for(i=0; i<=400000; i++) // Booster Loop
        {
            DDRB |= (1<<DDB2); // Open Drain
            DDRB &= ~(1<<DDB2); // Pull down
        }
        PORTA &= ~(1<<PA7); // LED OFF
        _delay_ms(500);
    }
}
```

Experimentelle Ultraschall- Waschmaschine

Von **Andrey M. Shustov** und **Michael A. Shustov**

Durch den Einsatz fortschrittlicher energiesparender Technik konnte eine Ultraschall-Waschmaschine für den Hausgebrauch entwickelt werden. Das Waschen erfolgt durch die Verdichtungs- und Verdünnungswellen, die periodisch vom Ultraschallgeber in einem praktisch inkompressiblen Medium wie Wasser verursacht werden. Kleidung und andere zu reinigende Gegenstände, die in einer solchen Flüssigkeit schwimmen, sind intensiven hydroakustischen Effekten ausgesetzt. Durch hydroakustische Wellen treten ultramikroskopische Gasblasen auf, die zur Abscheidung von Schmutzmikropartikeln aus der schmutzigen Kleidung oder anderen zu reinigenden Objekten beitragen. Die Bildung und der anschließende Kollaps der Gasblasen führt zudem zur Bildung von Ozon, das bekanntermaßen zur Sterilisierung von Kleidung beiträgt.

Der Vorteil des Ultraschall-Waschens besteht darin, dass dabei die Kleidung nicht deformiert oder gequetscht wird; selbst Woll- und Seidenkleidung und dünne Wäsche kann sicher gewaschen werden. Aber das Verfahren ist nicht nur zum Waschen und Desinfizieren von Kleidung geeignet, auch Gemüse und Obst für die Konservenherstellung lässt sich so behandeln. Ja, selbst Wasser kann man auf diese Weise desinfizieren!

Die Schaltung in **Bild 1** für ein experimentelles Ultraschall-Waschgerät mittlerer Leistung besteht aus einer Stromversorgungsquelle (IC2), zwei miteinander verbundenen Oszillatoren, die auf Frequenzen von 10 kHz und 1 MHz schwingen (IC1), einer Ausgangsstufe in Form eines Leistungstransistors T1 und einem Ultraschallwandler, der mit den Punkten C und D verbunden wird.

Die Stromversorgungsquelle im Prototyp ist nicht geregelt und für eine maximale Leistungsaufnahme von etwa 3 W ausgelegt, was ausreichen sollte, um Wäsche in einem Flüssigkeitsvolumen von 10...25 Litern zu waschen. Es ist sinnvoll, eine Ultraschall-Waschmaschine, auch diese experimentelle, mit einer Steuerung zu versehen, die eine stufenlose Einstellung der Ausgangsleistung ermöglicht. IC2 arbeitet als Konstantstromquelle, die sich mit P2 im Bereich von 25...1000 mA einstellen lässt. Es gibt in **Bild 2** auch eine alternative Schaltung dafür, eine einstellbare Konstantspannungsquelle mit einem Bereich von 5...13 V, die zwischen den Punkten A und B angeschlossen wird.

Der Pulspaket-Oszillator ist mit einem Standard-CMOS-IC CD4011 aufgebaut und hat keine besonders bemerkenswerten Eigenschaften. Um die Oszillatorfrequenz an die Resonanzfrequenz des Ultraschallsenders anzupassen, ist es notwendig, die Werte des RC-Glieds im Hochfrequenzoszillator (R4 und C3 bei IC1.3/IC1.4) zu optimieren. Sowohl der Regler IC2 als auch der Transistor T1 sollten mit einem Kühlkörper ausgestattet sein.

Achten Sie darauf, einen Ultraschallwandler zu wählen, der wasserdicht ist oder gemacht werden kann, denn nur wenn dieser ins Reinigungsbad eintaucht, kann er Ultraschallschwingungen mit maximaler Energie in die Umgebung (die Flüssigkeit) leiten. Als Ultraschallsender

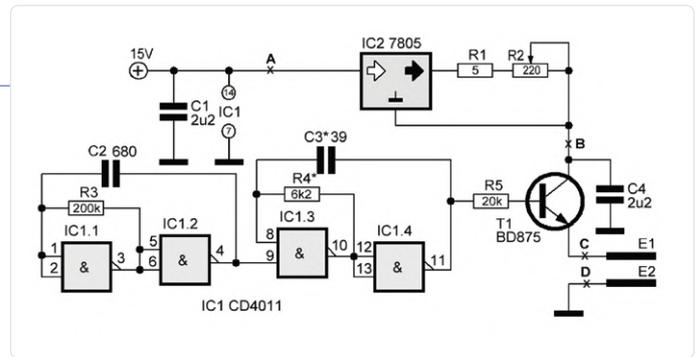


Bild 1. Schaltung einer experimentellen Ultraschall-Waschmaschine.

werden gewöhnlich piezoelektrische Keramiken (Bariumtitanat und Strontiumtitanat), Wandler mit Ferrit- oder Permalloy-Kern oder auch piezoelektrische Platten verwendet.

Das Projekt reizt natürlich zu weiteren Experimenten. So könnten Sie Ultraschallschwingungen auch erzeugen, indem Sie mit Hilfe eines Paares eng beieinander liegender Elektroden an den Punkten A und B elektrische Stromimpulse durch das Wasser leiten. Diese Stromimpulse zwischen den Elektroden bewirkt eine akustische, elektrostimierte Modulation. Als Elektrodenmaterial empfiehlt sich hier Aluminium. Die Schaltung muss, selbst wenn sie experimentell ist, zuverlässig von der Netzspannung isoliert sein. Der Waschbehälter sollte sich auf einem trockenen Fußboden in einem gewissen Abstand von geerdeten Gegenständen befinden.

Beim Wäschewaschen in einer Ultraschall-Waschmaschine geht man wie folgt vor:

- Das Waschpulver wird in das Waschwasser gegeben, wobei die gleiche Dosis wie bei der Handwäsche verwendet wird
- die Wassertemperatur sollte etwa 65°C betragen, aber probieren Sie es auch bei niedrigeren Temperaturen
- die Wäsche sollte in der Lösung frei schwimmen und gelegentlich mit einem Holzlöffel umgerührt werden.

Stark verschmutzte Bereiche der Kleidung sollten zusätzlich vorbehandelt werden. Der Waschvorgang dauert je nach Leistung und Effizienz des Ultraschallwandlers mindestens 30...40 Minuten. ◀

200237-03

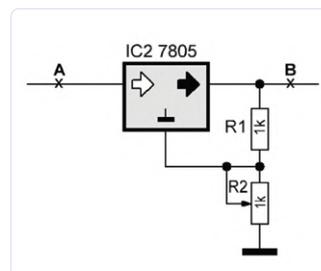


Bild 2. Einstellbare Spannungsquelle für die Ultraschall-Waschmaschine.

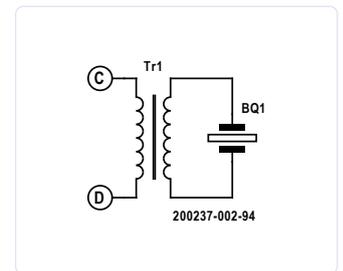


Bild 3. Schaltung des Ultraschallsenders. Das Gerät und seine Anschlüsse müssen absolut wasserdicht sein!



IM ELEKTOR-STORE

➤ Buch „Electronic Circuits For All“
www.elektor.de/electronic-circuits-for-all



Signalgenerator Joy-iT JDS2915

Dual-Signalgenerator mit Frequenzzähler im Metallgehäuse

Von **Dr. Thomas Scherer**

Von Joy-iT gibt es jetzt mit dem Modell JDS2915 für unter 100 € eine preiswertere Ausführung des Signalgenerators JDS6600, der statt bis 60 MHz nur Signale bis 15 MHz generieren kann – was aber für viele Fälle ausreicht.

Letzten Frühling hatte Kollege Harry Baggen mit dem JDS6600 den großen Bruder getestet. Dort hat ihn vor allem das etwas einfache Plastikgehäuse gestört. Nun wurde mit dem Modell JDS2915 eine kleinere Version in einem stabilen Alu-Profilgehäuse aufgelegt. Harry wird schmunzeln, da seine Gebete zum heiligen Electricus wohl erhört wurden. Da sein getestetes Modell im Prinzip gleich funktioniert und sich nur durch ein anderes Gehäuse, einen niedrigeren Preis und eine höhere Frequenzgrenze (meines hat „nur“ 15 MHz) unterscheidet, versuche ich, Redundanzen zu vermeiden. Man lese daher erst einmal das JDS6600-Review [1] von Harry. Ich werde mich anderen Aspekten widmen.

Auspacken

Bild 1 zeigt den kompletten Lieferumfang: Neben dem eigentlichen Signalgenerator im formschönen Metallgehäuse gibt es zwei BNC-Kabel mit Krokodklemmen für die zwei Signalausgänge, ein kurzes Kabel mit BNC-Steckern an den Enden zur Verbindung

mit anderen Geräten, ein USB-2.0-Kabel mit USB-A und USB-Stecker zur Steuerung des Signalgenerators via PC, ein Steckernetzteil mit Hohlstecker (5 V / 2 A) und eine dünne Anleitung im DIN-A5-Format.

Wer jetzt gedacht hat: „Ein Handbuch! Endlich mal ein richtiges Handbuch!“, der hat sich zu früh gefreut, denn sinnvolle Infos sind pro Sprache (Deutsch, Französisch und Englisch) auf jeweils nur zwei Seiten enthalten. Das richtige Handbuch findet sich auf der Webseite [2] von Joy-iT (in Deutsch und Englisch). Dort kann man auch ein Datenblatt und die PC-Software zur Fernsteuerung des Signalgenerators (samt Anleitung) downloaden.

Im dünnen DIN-A5-Heft wird lediglich (wie in **Bild 2**) erläutert, welche Knöpfe was bedeuten. Nicht so optimal am hübschen Alu-Profilgehäuse ist (in meinen Augen), dass die Ein- und Ausgänge rechts an der Seite angebracht sind und nicht wie beim JDS6600 vorne. Die Bedienung unterscheidet sich in praktisch nichts. Lediglich die „Control Keys“ sitzen beim JDS2915 etwas tiefer

und darüber direkt der Drehgeber. Beim JDS6600 musste dieser Drehknopf nach rechts ausweichen – was aber letztlich egal ist. Ich hatte schon vor dem Ausprobieren Bedenken, dass man das hochkant gestellte Gehäuse (Display nach vorne) mit einer Hand festhalten muss, wenn man eine Taste mit der anderen Hand betätigen will. Ich habe mich nicht getäuscht. Es ist leicht und rutscht weg, auch weil es keine Gummifüße hat.

Daten und Fähigkeiten

Zunächst die Standard-Angaben, da die im Datenblatt nicht immer ganz vollständig sind:

- › Versorgung: Euro-Steckernetzteil mit 5 V und 2 A
- › 2 Signalausgänge mit 50 Ω Impedanz
- › Ausgangsspannung ≤ 10 MHz: 0..20 V_{SS} , in 1-mV-Schritten
- › Ausgangsspannung > 10 MHz: 0..10 V_{SS} , in 1-mV-Schritten
- › Offset: -10 – +10 V in 10-mV-Schritten
- › Kurvenformen: Sinus, Rechteck, Dreieck, Impuls, arbiträr etc.
- › Frequenz Signale: 0 – 15 MHz, in 10-mHz-Schritten
- › Tastverhältnis (Impuls und Dreieck): 0,0..99,9 %
- › Spezialfunktionen: Sweeps für Sinus-Signale und Impulse
- › Frequenzzähler: 0 – 100 MHz
- › Frequenzgenauigkeit: ± 22 ppm
- › Frequenzstabilität: ± 1 ppm/3 h
- › Amplitudenstabilität: ± 5 %/5 h
- › Digitale Auflösung der Signale: 14 Bit
- › Abtastrate der Signale: 266 MS/s
- › Abmessungen: 145x95x55 mm (BxHxT)
- › Gewicht: 450 g (ohne Netzteil)
- › Stromaufnahme an 5 V: max. 850 mA (gemessen)
- › Display: 2,4" Farb-LCD
- › Bedienung: Tastenfeld, Drehgeber, WLAN, USB via PC-App

Anders als im Datenblatt angegeben kann der Signalgenerator auch Impulse bis 15 MHz liefern. Die Ausgangsstufe kann nicht viel mehr als ± 10 V ausgeben. Erhöht man den Offset z.B. auf das positive Maximum von +9,99 V, wird das Ausgangssignal auf 200 m V_{SS} begrenzt. Bei einem Offset von z.B. -4 V bleibt so noch Raum für Signale von maximal 12 V_{SS} . Wissenswert, aber in der Praxis keine wirkliche Einschränkung.

Ausprobieren

Eigentlich ist die Bedienung des Signalgenerators (abgesehen von den arbiträren Signalen) weitestgehend selbsterklärend. Ich kam jedenfalls auf Anhieb damit zurecht, ohne einen Blick ins PDF-Handbuch [3] zu werfen. Die momentan nur in Deutsch vorliegende Software-Anleitung [4] zeigt einen Signalgenerator im „normalen“ Kunststoffgehäuse: Ein Hinweis darauf, dass es sich bei der JDS6600er-Serie und dem JDS2915 um technisch weitestgehend gleiche Geräte handelt, was schon aufgrund der fast identischen Front meine Vermutung war. Die zugehörige Software ist eigentlich für das Modell JDS6600 gemacht – das ist der Beweis.

Die Software selbst ist ein mit LabView erstelltes Programm (Bild 3). Diese „Programmiersprache der 4. Generation“ von National

Bild 3. Die Optik des PC-Programms zur Fernsteuerung des JDS2915 verrät, dass es mit LabView gestrickt ist.



Bild 1. Der Lieferumfang beim Signalgenerator JDS2915.

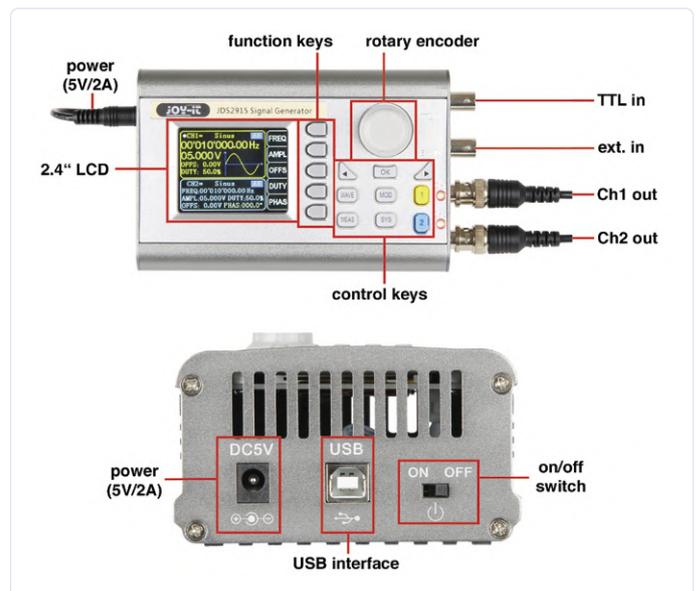


Bild 2. Das sind die Buchsen, Taster und sonstige Bedienelemente beim JDS2915.

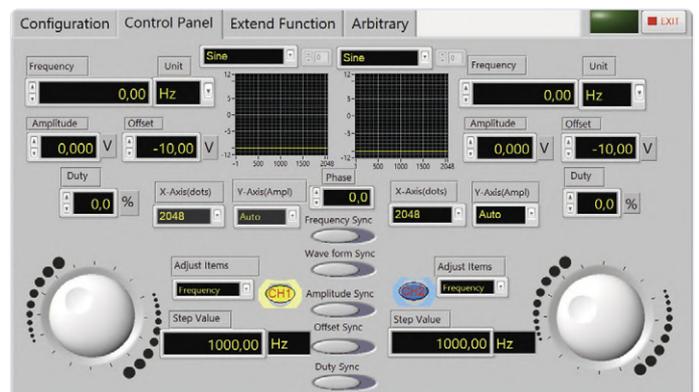




Bild 4. Bildschirm bei drei verschiedenen Modi: Sinus (links), Dreieck (Mitte) und Frequenzmessung (rechts).

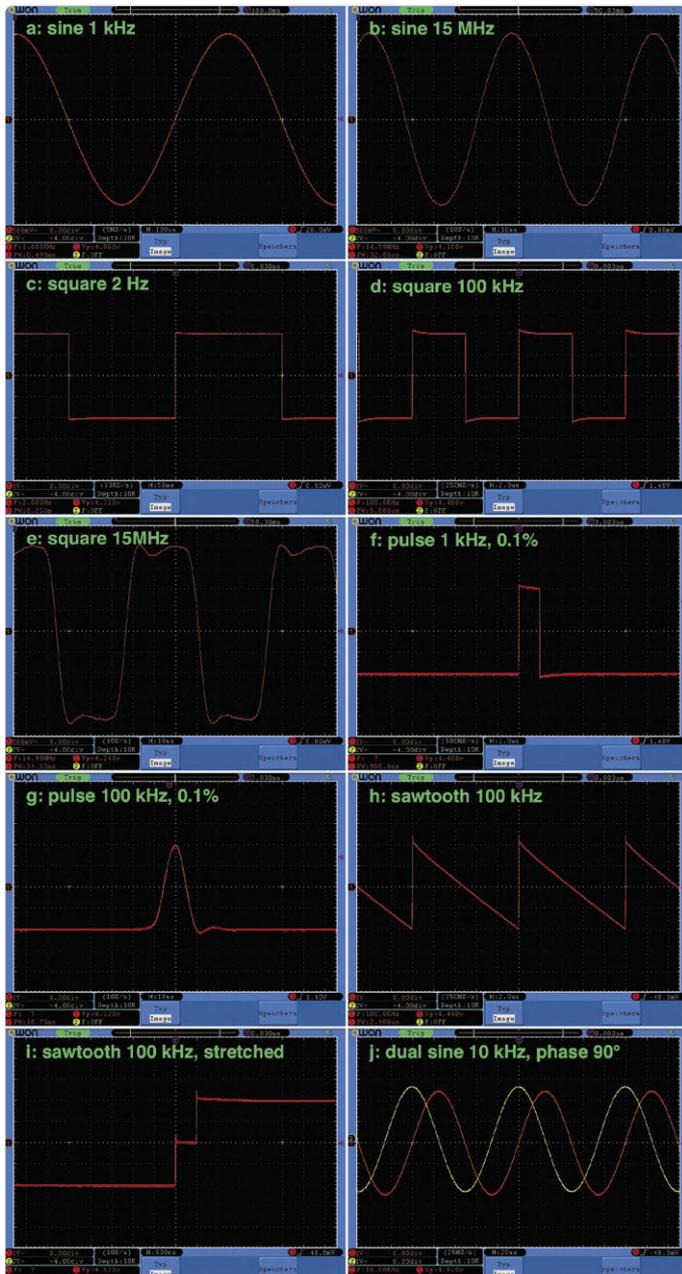


Bild 5. Diverse Signale mit $4 V_{SS}$: 5a: Sinus mit 1 kHz; 5b: Sinus mit 15 MHz; 5c: Rechteck mit 2 Hz; 5d: Rechteck mit 100 kHz; 5e: Rechteck mit 15 MHz; Impulse mit einem Tastverhältnis von 0,1%: 5f: 1 kHz und 5g: 100 kHz; 5h: Sägezahn bzw. Dreieck mit 1 kHz und einem Tastverhältnis von 0,1%; 5i: voriges Signal stark gestreckt; 5j: zwei 10-kHz-Sinus-Signale mit einer Phasenverschiebung von 90° .

Instruments wird häufiger beim professionellen Testen, Messen und Steuern eingesetzt – also ein gutes Zeichen. Obwohl bei Joy-iT (oder dem Hersteller der Software) die Treiber zur Anbindung des Signalgenerators zur Verfügung stehen, werden sie nicht zum Download angeboten. Deshalb kann man den Signalgenerator leider nicht in automatisierte Testumgebungen unter LabView einbinden. Das ist echt schade. Aber vielleicht liest das ja jemand bei Joy-iT. Wäre echt ein cooles Feature.

Bedienung und Signale

Bild 4 zeigt das Display in drei verschiedenen Modi (es gibt mehr). Oben ist die Einstellung für ein Sinus-Signal mit $4 V_{SS}$ bei 1 kHz zu sehen. Der Offset beträgt $+1,0 V$ und „DUTY“ aka Tastverhältnis wird hier und beim Rechteck ignoriert. Der blaue Kanal 2 steht auf 10 kHz bei $5 V_{SS}$ und 0 V Offset. Interessant ist hier der Parameter „PHAS“ = Phase, denn hier kann man eine Phasenverschiebung zwischen den beiden Signalen von 0 bis 360° einstellen.

Bei den Impulsen macht das Tastverhältnis Sinn, aber interessant ist dieses beim Dreieck (Mitte). Ein Dreieck zeigt sich nämlich nur bei einem Tastverhältnis von 50 %. Weniger biegt die Spitze nach links und mehr nach rechts. Bei sehr kleinen (wie abgebildet) bzw. sehr großen Werten ergibt sich ein Sägezahn.

Der Frequenzzähler (rechts) zeigt sich beim Druck auf den Taster „MEAS“. Hier habe ich das 1-kHz-Kalibrationssignal meines Oszilloskops erfasst. Die Anzeige mäandert zwischen 999 Hz und 1 kHz hin und her. Anscheinend ist dieses Signal nicht perfekt symmetrisch.

Das Wichtigste an einem Signalgenerator sind die Qualität und die Art der erzeugten Signale. **Bild 5** gibt hiervon einen Eindruck. Die Signale haben eine eingestellte Amplitude von $4 V_{SS}$ und wurden mit meinem 100-MHz-Oszilloskop gemessen, das einkanalig eine Abtastung mit vollen 1 GS/s erlaubt.

Sinus

Der Sinus von **Bild 5a** sieht sehr sauber aus. Schaut man ihn sich genau an, zeigen sich kleine Treppchen. Doch falscher Alarm: An diesen Stellen ist der 8-Bit-A/D-Wandler meines Oszilloskops schuld. Real sind sie dank der 14 Bit des Generators viel feiner aufgelöst. Ich habe das per Dehnung der X- und Y-Ablenkung überprüft: Die Treppchen zeigen sich erst bei 100-facher Vergrößerung! Auch bei 15 MHz sieht das Signal von **Bild 5b** noch relativ proper aus und lediglich etwas verzerrt. Es zeigt sich fast keine Abschwächung des Signals.

Rechteck

In **Bild 5c** ist ein Rechtecksignal bei 2 Hz (DC-gekoppelt) zu sehen. Die angezeigten 4,12 V liegen an der Ungenauigkeit meines Oszilloskops. Tatsächlich sind es 4,03 V! Bei 100 kHz zeigen sich in **Bild 5d** leichte Überschwinger am Ende der positiven und negativen Flanke – nicht schön, aber akzeptabel. Bei 15 MHz in **Bild 5e** machen sich Schwächen des analogen Ausgangsverstärkers bemerkbar – immerhin erkennt man noch etwas Rechteckmäßiges.

Impuls

Der Nadelimpuls mit einem Tastverhältnis von 0,1% bei 1 kHz in **Bild 5f** hat mit 980 ns Breite einen Fehler von 2 %. Interessant wird es bei höheren Frequenzen: Oberhalb von 100 kHz wird der kürzeste Impuls bei eingestellten 0,1% nicht mehr schmaler als etwa 10 ns (in Pegelmitte, **Bild 5g**). Treibt man die Frequenz über 1 MHz, wird nur die Amplitude geringer und sinkt bei >10 MHz auf etwa 2 V. Das ist kein Wunder, denn die Abtastrate des Generators von 266 MHz erlaubt ja nur Schritte bis knapp unter 4 ns. Anders als laut Datenblatt kann man Impulse bis 15 MHz einstellen.

Dreieck

Bild 5h zeigt ein 100-kHz-Dreiecksignal, das bei „DUTY“ von 0,1% zu einem linkslastigen Sägezahn mutiert. An der positiven Flanke zeigen sich leichte Überschwinger. Dehnt man die Horizontalablenkung um das Fünffache (**Bild 5i**), kann man eine Abtaststufe sehen. Bei einem Tastverhältnis von 0,0% verschwindet diese Stufe komplett und ein toller Sägezahn zeigt sich.

Phase

Weil das schön und nützlich ist, wenn man zwei Signale zugleich generieren kann, wird in **Bild 5j** schließlich der Effekt von zwei um 90° gegeneinander phasenverschobenen 10-kHz-Sinus-Signalen (ausnahmsweise mit 5 V_{SS}) demonstriert.

Fazit

Das Schlechte am Guten ist aus meiner Sicht das Gehäuse. Klar ist ein Alu-Profilgehäuse technisch besser als eines aus Kunststoff. Aber für mich ist das Gerät mit seinen 450 g zu leicht, vor allem verglichen mit meinem alten teleanalogen Funktionsgenerator (**Bild 6**), der gut 2,5 kg wiegt. Dort rutscht nichts, wenn ich die Tasten einhändig betätige. Auch mag ich es lieber, wenn die BNC-Buchsen an der Front angebracht sind – aber Bedürfnisse sind unterschiedlich und vielleicht mögen Sie das Modell JDS2915 ja genau deswegen.

Technisch gibt es an diesem Signalgenerator mit eingebautem Frequenzzähler kaum etwas zu mäkeln. Die Signale sind überraschend gut - sehr viel besser als bei meiner alten Möhre. Mehr als 10 MHz brauche ich sowieso so gut wie nie und die Fernsteu-



Bild 6. Zum Vergleich: So sieht mein alter, teleanaloger Funktionsgenerator aus. Ob da noch ein XR2206 drin steckt?

erung per PC ist einfach super. Ein LabView-Treiber ist bei dem Preis nicht wirklich erwartbar. Überhaupt: für knapp unter 100 € (für Elektor-Mitglieder) ist dieser Funktionsgenerator ein Preis/Leistungs-Hammer! ❏

200143-01



IM ELEKTOR-SHOP

> Signalgenerator Joy-iT JDS2915

www.elektor.de/joy-it-jds2915-signal-generator-15-mhz



WEBLINKS

- [1] Review des Modells JDS6600: www.elektormagazine.de/news/review-dds-signalgenerator-joy-it-jds6600
- [2] Webseite von Joy-iT: <https://joy-it.net/de/products/JT-JDS2915>
- [3] Handbuch (PDF): <https://joy-it.net/files/files/Produkte/JT-JDS2915/JT-JDS2915-Anleitung-07.02.20.pdf>
- [4] Software-Anleitung (PDF): <https://joy-it.net/files/files/Produkte/JT-JD6600/JT-JDS-Software-Anleitung.pdf>

Ampelsteuerung in PIC-Assembler

Von **Andrew Pratt**

Dieses kleine Projekt zeigt, wie die Steuerung einer Ampelanlage programmiert werden kann. Es handelt sich um zwei Ampeln, wie man sie zum Beispiel an (englischen) Baustellen vorfindet, an denen sie den Verkehr auf nur einer Fahrspur freigeben beziehungsweise sperren (**Bild 1**). Die Ampeln werden von je einer roten, einer gelben und einer grünen LED symbolisiert. Die Programmierung erfolgt in PIC-Assembler und nicht in einer höheren Programmiersprache.

Beim Einschalten sind beide Ampeln rot und verbleiben für 10 s in diesem Zustand. Dann wechselt Ampel A für 2 s auf rot/gelb und anschließend auf grün. Nach weiteren 20 s wird Ampel A für 5 s gelb und dann rot. Beide Ampeln bleiben dann 10 s rot, bevor Ampel B den gleichen Zyklus durchläuft. Das alles wiederholt sich, bis der Strom abgeschaltet wird. Zusätzlich gibt es ein Enable-Signal, das die Lichtsequenz aktiviert. Wenn dieses Signal low wird, stoppt die Sequenz beim nächsten Doppel-Rot. Das Freigabesignal muss während der gesamten 10-s-Periode high sein, in der beide Ampeln vor dem Start der Sequenz rot sind.

Folgende Tabelle macht die verschiedenen Zustände, in denen sich das System befinden kann, mehr als deutlich:

Status	Ampel A	Ampel B
0	Rot	Rot
1	Rot - Gelb	Rot
2	Grün	Rot
3	Gelb	Rot
4	Rot	Rot
5	Rot	Rot-Gelb
6	Rot	Grün
7	Rot	Gelb

Bild 2 zeigt die Steuerung mit den acht verschiedenen Zuständen und den Eingangsbedingungen (für den zulässigen Übergang zum nächsten Status) als Moore-Diagramm [1]. Dies ist insofern ein sehr einfaches Beispiel, als die Zustände nacheinander mit nur einem einzigen zulässigen Übergang zum nächsten Zustand aufeinander folgen. Die Ausgänge hängen nur vom aktuellen Zustand ab und sind im Diagramm innerhalb der Box für jeden Zustand dargestellt.

Der Übergang in den nächsten Zustand wird durch die verstrichene Zeit in einem bestimmten Zustand bestimmt, mit Ausnahme der Ausgangszustände 0 und 4, in denen beide Ampeln rot sind. In diesen Status muss für 10 s auch das Freigabesignal (Enable) anliegen. Im Schaltplan in **Bild 3** wird das Enable-Signal durch einen Tastendruck erzeugt. Die sechs LEDs sind über Strombegrenzungswiderstände angeschlossen. Der Wert dieser Widerstände ist nicht kritisch, 1 k Ω ergibt etwa 3 mA. Es wird ein PIC des Typs 16F823 verwendet, der über eine FTDI-USB/Seriell-Brücke programmiert wird. Die am USB-Verbinder genannten

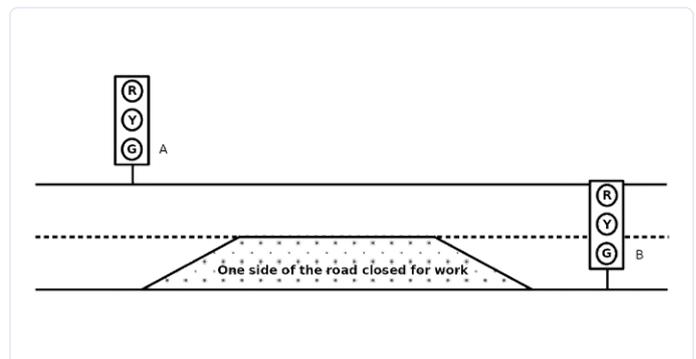


Bild 1. Verkehrssteuerung an einer (englischen!) Baustelle.

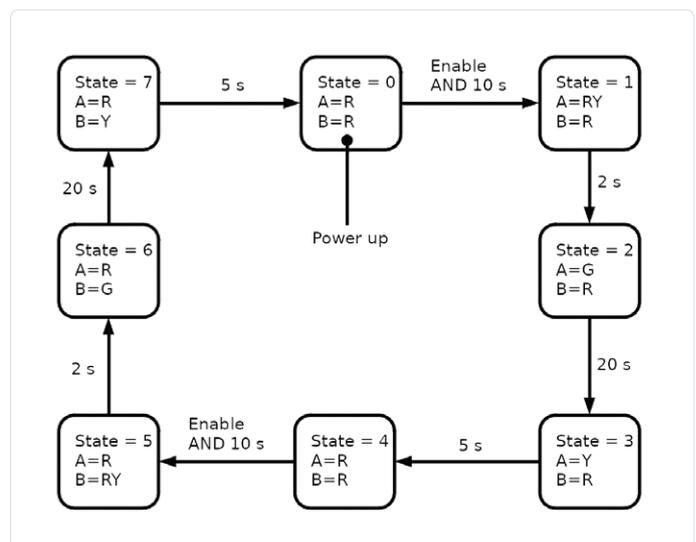


Bild 2. Zustandsdiagramm für den Straßenbau.

Listing 1. Vollständiges Programm für die Ampelsteuerung.

```
;PROG_8_03.asm
LIST      P=16F1823
#include <p16f1823.inc>
#include <fsm_macros.inc>

RADIX DEC          ; Default numbers are to base 10.
BOOK_CONFIGURATION ; See macro in fsm_macros.inc.

CBLOCK 0x70
TICKS          ; Used to count 8.2ms ticks.
SECONDS        ; Used to hold the seconds count.
ENDC

ORG 0x00
GOTO START

ORG 0x04          ; Interrupt vector.
BCF INTCON, TMR0IF ; Clear the tmr0 overflow interrupt flag.
DECF TICKS, F     ; Decrement by one the 8.2ms ticks.
BTFSS STATUS, Z   ; Test to see if TICKS has reached zero.
GOTO $+4          ; If TICKS has not reached zero jump to return from interrupt.
INCF SECONDS, F   ; If TICKS has reached zero increment the seconds count.
MOVLW 122
MOVWF TICKS      ; Restore TICKS to 122 (122 x 8.2 ms 1 second).
RETFIE          ; Return from interrupt.

START
MOVLB 1          ; Bank 1 required for the following macros, and TRISC.
SET_FREQ_32MHZ  ; See file fsm_macros.inc.
SET_TMR0_CASE1  ; Gives 8.2 ms ticks. See fsm_macros.inc.
CLRF TRISC      ; Set all PORTC as outputs.
MOVLB 0          ; Select bank 0 for PORTA and PORTC.
;=====
S0
MOVLW b'00100100'
MOVWF PORTC      ; Turn Red A (RC5) and Red B (RC2) on others off.
SS0_0
CLRF SECONDS     ; Clear the seconds counter.
SS0_1
BTFSS PORTA, 5   ; If the enable signal is low return to sub-state 0.
GOTO SS0_0
MOVLW 10
IF_REG_LESS_THAN_W SECONDS
GOTO SS0_1       ; If less than 10 s have elapsed stay in substate 1.
;=====
S1
MOVLW b'00110100'
MOVWF PORTC      ; Turn Red A (RC5) and Yellow A (RC4) and Red B (RC2) on others off.
SS1_0
CLRF SECONDS     ; Clear the seconds counter.
SS1_1
MOVLW 2
IF_REG_LESS_THAN_W SECONDS
GOTO SS1_1
S2
MOVLW b'00001100'
MOVWF PORTC      ; Turn Green A (RC3) and Red B (RC2) on others off.
SS2_0
CLRF SECONDS
SS2_1
MOVLW 20
IF_REG_LESS_THAN_W SECONDS
GOTO SS2_1       ; If less than 20 s have elapsed stay in substate 1.
```

Forts. nächste Seite...

```

;=====
S3
    MOVLW b'00010100'
    MOVWF PORTC                ; Turn Yellow A (RC4) and Red B (RC2) on others off.
SS3_0
    CLRF SECONDS                ; Clear the seconds counter.
SS3_1
    MOVLW 5
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS3_1                ; If less than 5 s have elapsed stay in substate 1.
;=====
S4
    MOVLW b'00100100'
    MOVWF PORTC                ; Turn Red A (RC5) and Red B (RC2) on others off.
SS4_0
    CLRF SECONDS                ; Clear the seconds counter.
SS4_1
    BTFSS PORTA, 5
    GOTO SS4_0                ; If the enable signal is low return to sub-state 0.
    MOVLW 10
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS4_1                ; If less than 10 s have elapsed stay in substate 1
;=====
S5
    MOVLW b'00100110'
    MOVWF PORTC                ; Turn Red A (RC5) and Red B (RC2) and Yellow B (RC1) on others off.
SS5_0
    CLRF SECONDS                ; Clear the seconds counter.
SS5_1
    MOVLW 2
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS5_1                ; If less than 2 s have elapsed stay in substate 1 .
;=====
S6
    MOVLW b'00100001'
    MOVWF PORTC                ; Turn Red A (RC5) and and Green B (RC0) on others off.
SS6_0
    CLRF SECONDS                ; Clear the seconds counter.
SS6_1
    MOVLW 20
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS6_1                ; If less than 20 s have elapsed stay in substate 1.
;=====
S7
    MOVLW b'00100010'
    MOVWF PORTC                ; Turn Red A (RC5) Yellow B (RC1) on others off.
SS7_0
    CLRF SECONDS                ; Clear the seconds counter.
SS7_1
    MOVLW 5
    IF_REG_LESS_THAN_W SECONDS
    GOTO SS7_1                ; If less than 5 s have elapsed stay in substate 1.
    GOTO S0

END

```

Farben entsprechen denen des entsprechenden FTDI-Kabels. Vor der Erzeugung des Codes, mit dem die obige Zustandsmaschine läuft, müssen am PIC bestimmte Konfigurationen in der Konfigurationsdatei, bei der Oszillatorfrequenz und den Anschlüssen vorgenommen werden. Es ist am besten, bei der Programmentwicklung schrittweise vorzugehen und stets mit Hilfe eines Debug-Codes zu testen. So können Sie sicherstellen, dass jedes Bit, das Sie hinzufügen, auch tatsächlich funktioniert. Ein komplettes Programm „am Stück“

zu schreiben und dann zu versuchen, es zum Laufen zu bringen, ist dagegen nicht der richtige Weg.

Die Top-Level-State-Machine des Assemblercodes ist in **Bild 4** dargestellt, der eigentliche PIC-Assemblercode ist in **Listing 1** zu sehen. Die Besprechung des Programmlistings und der damit verbundenen Komponenten zur Zeitmessung und Ein-/Ausgabe-Prüfung sprengen leider den Rahmen dieses kleinen Artikels, und dasselbe gilt für ein spezielles Test-Programm und einen Mini-Debugger für den internen

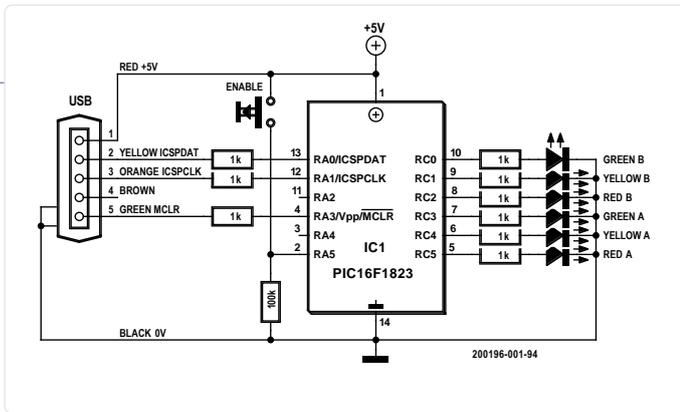


Bild 3. Schaltplan der (virtuellen) Ampelsteuerung.

Sekundenzähler. Die beiden letzteren sind aber ziemlich hilfreich, um das Programm schrittweise und in einer lehrreichen Weise aufzubauen und können deshalb wie alle (Sub-)Routinen und die zugehörigen Zustandsdiagramme von [2] kostenlos heruntergeladen werden. ◀

200196-02

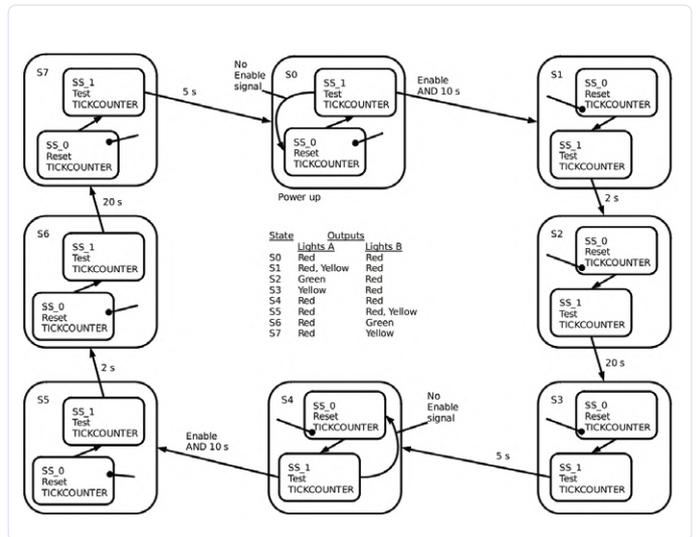


Bild 4. Komplettes Zustandsdiagramm der Ampelsteuerung.

WEBLINKS

- [1] **Moore-Automat:**
<https://de.wikipedia.org/wiki/Moore-Automat>
- [2] **Download zum Projekt:**
www.elektormagazine.de/200196-02



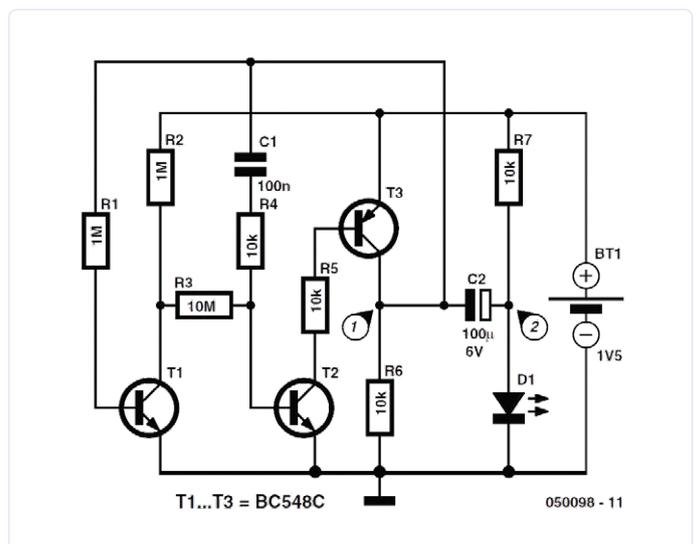
IM ELEKTOR-STORE

➤ **Buch „Programming Eight Bit PICs in Assembly as State Machines“:** In Kürze erhältlich, siehe www.elektor.de/books

Der ewige Blinker

Von **Burkhard Kainka**

In Geschäften sieht man manchmal Werbeschilder mit einer blinkenden LED, die anscheinend ewig blinken, und das auch noch mit einer einzelnen Batteriezelle. So etwas müsste man auch mal selbst bauen ... Hier ist die Schaltung. Sie besteht aus einem astabilen Multivibrator mit besonderen Eigenschaften. Ein Elko von 100 µF lädt sich relativ langsam und mit kleinem Strom auf und wird in einem kurzen Impuls über die LED entladen. Dabei kommt es auch zur nötigen Spannungsüberhöhung, denn 1,5 V ist ja für eine LED zu wenig. Die beiden Oszillogramme verdeutlichen, wie es funktioniert: Die Spannung am Kollektor des PNP-Transistors schaltet bis auf ca. 1,5 V hoch, nachdem der Elko an dieser Stelle über einen Widerstand von 10 k bis nahe 0,3 V entladen wurde. Auf der anderen Seite wurde er bis ca. 1,2 V geladen. Die Differenz von 0,9 V liegt also am Elko, wenn der Blinkimpuls erscheint. Sie addiert sich in diesem Moment zur Batteriespannung von 1,5 V, so dass die Impulshöhe an der LED bis 2,4 V betragen könnte. Tatsächlich aber zeigt Oszillogramm 2, dass die Spannung durch die LED auf ca. 1,8 V begrenzt wird. Die LED-Spannung passt sich selbst der verwendeten LED an und kann theoretisch bis an die 3 V betragen. Die Schaltung wurde für den Low-power-Betrieb optimiert. Deshalb ist das eigentliche Flip-Flop mit einem NPN- und einem PNP-Transistor aufgebaut. Man kann auf diese Weise die Verschwendung von

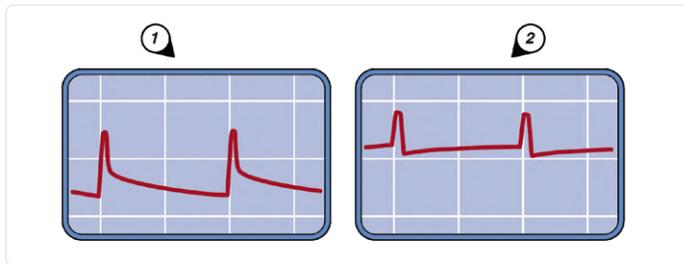


Steuerstrom vermeiden. Beide Transistoren leiten nur für den kurzen Moment des LED-Blinkens. Damit stabile Bedingungen herrschen und die Schaltung sicher schwingt, gibt es eine zusätzliche Stufe mit einer Gleichspannungs-Gegenkopplung. Auch hier wurde durch besonders hochohmige Widerstände auf geringsten Verbrauch geachtet.

Für die Abschätzung des Verbrauchs kann man vom Ladestrom des Elkos ausgehen. An den beiden Ladewiderständen mit je 10 k liegt im Schnitt eine Spannung von insgesamt 1 V. Damit beträgt der durchschnittliche Ladestrom 50 μ A. Für die Dauer des LED-Impulses wird noch einmal genau die gleiche Ladung aus der Batterie entnommen. Der mittlere Strom beträgt also rund 100 μ A. Geht man von einer Batteriekapazität von 2500 mAh aus, sollte die Batterie etwa 25.000 Stunden halten. Das sind über zwei Jahre, also (fast) ewig. Da der Strom mit absinkender Batterie-

spannung etwas abnimmt und die LED nicht mehr so hell leuchtet, dürfte die tatsächliche Betriebsdauer noch höher liegen. Also mehr als (fast) ewig. Fühlen Sie sich herausgefordert? Zeigen Sie uns Ihr Ergebnis auf www.elektormagazine.de/labs! ◀

200200-04



IM ELEKTOR-STORE

- > Buch „Elektronik-Grundlagen und Einsteiger-Projekte“
www.elektor.de/elektronik-grundlagen-und-einsteiger-projekte
- > Buch als PDF „Elektronik-Grundlagen und Einsteiger-Projekte“
www.elektor.de/elektronik-grundlagen-und-einsteiger-projekte-pdf

Experimenteller Hall-Sensor

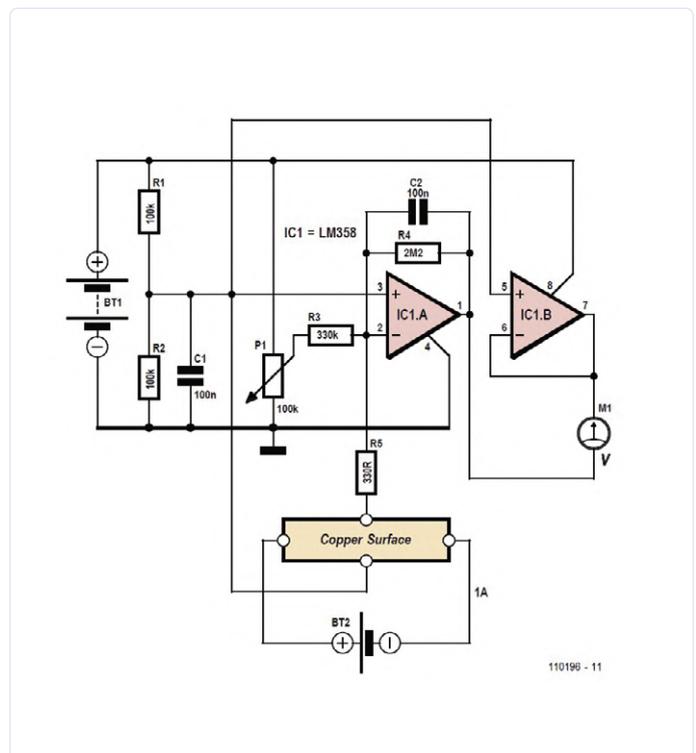
Von **Burkhard Kainka**

Hall-Sensoren kann man natürlich kaufen, aber selber bauen ist viel interessanter. Nach der Theorie kommt es darauf an, eine möglichst dünne Folie zu verwenden, die Länge und die Breite sind dagegen egal. Man kann es also auch mit Kupfer versuchen, weil es in Form von Platinenmaterial gut zu verarbeiten ist. Zwar ist Kupfer ein denkbar ungünstigstes Material, weil es eine sehr kleine Hall-Konstante hat. Aber es gibt ja inzwischen extrem starke Magnete, damit müsste man den Effekt auch bei einem Hall-Sensor aus Kupfer nachweisen können. Um den Nachweis zu erbringen, kommt es auf eine möglichst große Verstärkung an. Die Spannungsverstärkung der gezeigten Schaltung wird durch das Verhältnis der beiden Gegenkopplungswiderstände des ersten Opamps festgelegt. Mit den angegebenen Werten (2,2 M Ω und 330 Ω) ergibt sich eine 6.667-fache Verstärkung. Dazu kommt eine Brückenschaltung für das Messgerät. Das Poti dient zum Feinabgleich.

Man könnte also bei genauer Nullpunkteinstellung im Millivoltbereich messen und damit Hallspannungen von weit unter einem Mikrovolt auswerten. Letztlich lässt sich auf diese Weise die Flussdichte eines Magneten messen. Kupfer hat eine Hallkonstante von $A_H = -5,3 \cdot 10^{-11} \text{ m}^3/\text{C}$. Die Dicke der Kupferschicht ist $d = 35 \mu\text{m}$. Die Hallspannung beträgt dann:

$$U_H = A_H \cdot I \cdot B/d$$

Für $B = 1 \text{ T}$ und $I = 1 \text{ A}$ ergibt sich demnach eine Hallspannung von $U_H = 1,5 \mu\text{V}$. Das Ganze 6.667-fach verstärkt ergibt 10 mV. Die Schaltung hat also eine Empfindlichkeit von 10 mV pro Tesla. Allerdings ist es nicht ganz einfach, den Nullpunkt mit P1 zu justieren. Der Verstärker



hat seine eigene Stromversorgung in Form einer 9-V-Batterie (BT1). Für die Messung schließt man ein Labornetzteil mit einstellbarem Ausgangsstrom (BT2) an den Hallsensor (die Kupferfläche) an und stellt den Strom durch den Sensor auf genau 1 A ein. Dann muss der Nullpunkt noch einmal neu justiert werden.

Anschließend legt man einen starken Neodym-Magneten unter den Sensor. Die Ausgangsspannung der Schaltung wird sich jetzt tatsächlich um einige Millivolt ändern. Allerdings gibt es mehrere Effekte, die das Messergebnis beeinträchtigen können. Jede Bewegung des Magneten erzeugt eine Induktionsspannung in den Zuleitungsdrähten, die wesentlich größer ist als die Hallspannung. Man muss nach jeder Bewegung des Magneten immer etwas warten, bis die Messwerte

wieder stabil stehen. Außerdem gibt es bei so kleinen Messspannungen Probleme mit Thermospannungen, die auf Temperaturunterschiede zurückzuführen sind. Man darf sich nicht bewegen und sollte nach Möglichkeit auch noch die Luft anhalten... ◀

200198-04

Kurzschlussverfolgung mit dem Milliohm- oder ESR-Meter

Von **Paul Hetrelezi**

Da ein ESR-Messgerät ja eigentlich ein Messgerät für sehr niederohmige Widerstände ist, kann man mit ihm wunderbar Kurzschlüsse zwischen Leiterbahnen lokalisieren, insbesondere, wenn ein Bild des Platinenlayouts auf Papier oder am Bildschirm zur Verfügung steht. Man klemmt die eine Messspitze als fixen Referenzpunkt an eine Leiterbahn und bewegt die andere Messspitze auf dieser Leiterbahn auf diesen Referenzpunkt zu oder von ihm weg. Eine gute Idee ist es, eine Versorgungsleitung als Referenzpunkt zu verwenden und dann eine Messspitze am Massepunkt (GND) zu platzieren. **Bild 1** zeigt zwei Stromversorgungsleitungen, die von der gemeinsamen (Gleichrichter- oder Schutz-) Diode und dem Filterkondensator abzweigen, eine zu Punkt A und die andere (in dieser Reihenfolge) zu den Punkten B, D und C. Angenommen, in der Schaltung liegt an einem dieser Punkte (hier Punkt C) ein Kurzschluss nach Masse vor, entweder an dem Zweig zu Punkt A oder an dem Zweig mit den Punkten A, B und C. Die bewegliche Messspitze wird langsam in

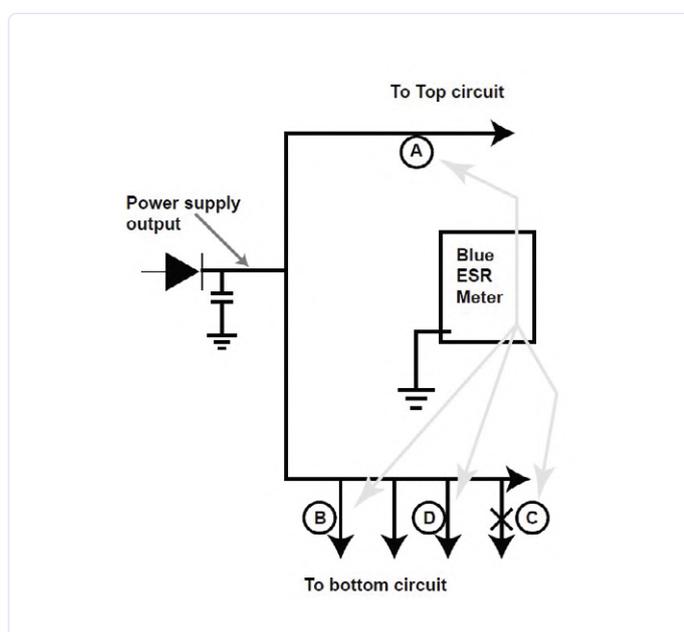
Richtung Punkt A wegbewegt. Dabei sollte sich, wenn alles in diesem Zweig in Ordnung ist, der Widerstand merklich erhöhen, da ja immer mehr Leiterbahnwiderstand zu überwinden ist.

Nun kehren Sie mit der Messspitze um und bewegen sie in Richtung der Punkte B, D und C. Sie sehen, dass der Messwert langsam abnimmt, wobei der niedrigste Messwert an Punkt C auftritt. Dies deutet auf einen Kurzschluss im Bereich C hin. Beachten Sie, dass diese Methode nur bei einseitigen Platinen funktioniert.

Tipps

- Führt man mit einem DMM einen Sperrschicht-Test in einer Schaltung durch, so sorgt ein über der Halbleiter-Sperrschicht parallel geschalteter Kondensator aufgrund der Lade- und Entladeeffekte für falsche Werte. Trotz der Mehrarbeit ist es deshalb am besten, solche „verdächtigen“ Bauteile auszulöten und außerhalb der Schaltung zu überprüfen.
- Bei der Messung des Widerstands eines Bauteils auf der Platine ist es ratsam, den Messwert von der Bauteilseite der Platine aus zu ermitteln, wo immer dies möglich ist. Dadurch wird sichergestellt, dass kein offener Stromkreis angezeigt wird, nur weil das geprüfte Bauteil eine kalte Lötstelle hat. Vergewissern Sie sich auch, dass sich der Messstrom nicht irgendwie parallel am zu messenden Bauteil vorbeimogeln kann! ◀

200212-03



IM ELEKTOR-STORE

- Buch „Retro Audio: A Good Service Guide“
www.elektor.de/retro-audio
- E-Buch „Retro Audio: A Good Service Guide“
www.elektor.de/retro-audio-e-book
- Peak Atlas ESR70 Plus ESR-Tester
www.elektor.de/peak-atlas-esr70-plus-capacitor-analyser
- Buch „Robert Lacoste's The Darker Side“
www.elektor.de/robert-lacoste-s-the-darker-side



Elektor SDR-Praxis-Bundle

Von **Harry Baggen**

Per „Software Defined Radio“ kann man mit einem Computer und relativ einfacher Hardware Funksignale empfangen und dekodieren. Das SDR-Praxis-Bundle von Elektor enthält alle wesentlichen Teile für diese moderne Form des Radioempfangs. Dieses Review gibt einen Eindruck von den Möglichkeiten des preiswerten Kits und von ersten Erfahrungen mit dem SDR.

In den letzten Jahren hat sich Elektor schon viel mit dem SDR beschäftigt. Es ging dabei nicht nur um die Prinzipien, sondern auch um praktische und leicht aufbaubare Hardware. 2007 erschien ein SDR-Empfänger in Form eines Shields für den Arduino Uno, bei dem letzterer für die Abstimmung des Oszillators auf der SDR-Platine sorgte. Das Ausgangssignal wurde an einen PC gesendet, der daraus mit Hilfe einer speziellen SDR-Software die Audiosignale destillierte. Damit konnten dann HF-Signale von 150 kHz bis 30 MHz empfangen werden.

Inzwischen sind zwei Nachfolger dieses beliebten SDR-Empfängers erschienen. Der letzte stammt von 2016. Für wenig Geld kann man eine Platine des Elektor SDR Shield 2.0 beziehen, und erhält so eine großartige Gelegenheit, die SDR-Technik näher kennenzulernen.

Kompletter Bausatz

Seit einiger Zeit bietet Elektor auch ein so genanntes SDR-Praxis-Bundle an, das ein Übungsbuch, ein aufgebautes SDR-Shield, Stifteleisten, zwei spezielle Ringkerne und ein Stück dünnes Koaxi-



Bild 1. Der Inhalt des Bundles.



Bild 2. Das wird noch benötigt: ein Arduino Uno, ein USB-Kabel, ein Kabel mit Klinkesteckern und ein Stück Draht.

alkabel enthält (**Bild 1**). Man braucht nur noch einen Arduino Uno, und die meisten von uns haben irgendwo ein solches Board rumliegen (**Bild 2**). Beim Buch handelt es sich um das SDR-Praxisbuch (erhältlich in Englisch und Deutsch), das von Burkhard Kainka, dem Entwickler des SDR-Shields, geschrieben wurde. Er erklärt auf anschauliche Weise, was man mit dem SDR-Shield anfangen kann. Er zeigt auch, wie man die verschiedenen SDR-Programme einsetzt, und beleuchtet auch das Thema Antenne, die für einen guten Empfang unerlässlich ist. Auch die Arduino-Software wird ausführlich besprochen. Schließlich werden einige größere Projekte behandelt, die auf dem SDR-Shield basieren: ein eigenständiger Empfänger, ein WSPR-Sender und ein QRP-Transceiver.

Eine Rezension des Buches [1] wurde bereits von Elektors HF-Spezialist Jan Buiting geschrieben. Seine Diskussion ist sehr ausführlich – Interessierten sei das Review wärmstens empfohlen! Ich selbst bin eher in der Audio- und Messtechnik zu Hause – aber gerade als Nicht-HF-Spezialist dürfte es doch besonders interessant sein, welche Erfahrungen ich mit diesem Kit machen würde.

Erste Schritte

Der erste Schritt bestand darin, die mitgelieferten Stiftleisten auf die Platine zu löten, die aus praktischen Gründen noch nicht vorbelegt sind. Außerdem habe ich eine zweipolige Stiftleiste an die Anschlüsse für den Antenneneingang gelötet. Damit kann man sehr einfach eine Antenne anstecken. Schön wäre es gewesen, solche Stiftleisten wären im Lieferumfang, denn nicht jeder hat solche in einer Schublade.

Als Nächstes wurde das Shield auf das Arduino-Board gesteckt. Jetzt kann der Uno an einen PC oder Laptop angeschlossen werden. Dazu wird ein USB-Kabel (USB A auf USB B) für die Daten und ein Kabel mit Stereo-3,5-mm-Klinkenstecker für die Audioverbindung benötigt. Spätestens jetzt sollte man das Buch zur Hand nehmen. Auf den ersten Seiten beschreibt der Autor kurz die Hardware des SDR-Shields, und dann wird es mit dem Programm G8JCFSDR direkt praktisch (**Bild 3**).

Der Download und die Installation des Programms ging schnell. Es unterstützt das Elektor-SDR-Shield komplett. Nach Auswahl des vorliegenden Shields im Konfigurationsmenü bietet das Programm die Möglichkeit, direkt die richtige Frequenzregelungssoftware in den Arduino zu laden, ohne die Arduino-IDE installieren zu müssen. Das lief alles ziemlich glatt. Bitte beachten Sie, dass Sie in G8JCFSDR unter *Display* die Option *full* anklicken müssen, sonst bleibt die Schaltfläche *Config* unsichtbar. Nachdem ich ein Stück Draht an den Antenneneingang angeschlossen hatte, konnte ich mich auf Sendersuche begeben. Ehe man sich's versieht ist ein Nachmittag vorbei und man hat alle Möglichkeiten und Einstellungen des Programms ausprobiert. Früher habe ich einmal mit einem einfachen Kurzwellenempfänger gespielt, aber das hier macht viel mehr Spaß und bietet unendlich viele Möglichkeiten! Das Programm G8JCFSDR bietet einen idealen Einstieg und das

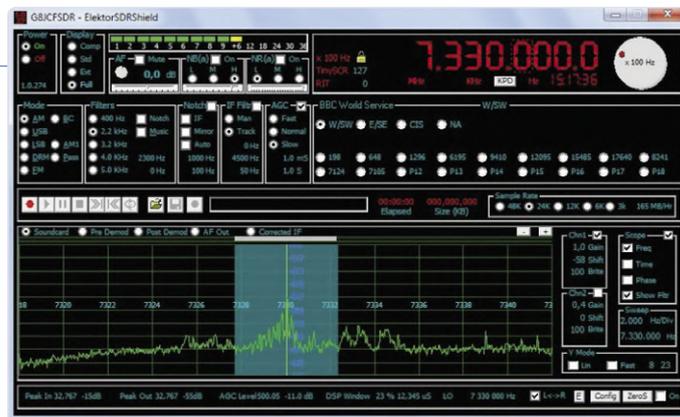


Bild 3. Das Programm G8JCFSDR bietet einen idealen Einstieg und das SDR-Shield wird voll unterstützt.

SDR-Shield wird voll unterstützt. Nach diesen ersten Erfahrungen konnte ich mit der Installation des Abstimmprogramms auf dem PC für den Oszillator auf dem SDR-Shield fortfahren (die gesamte vom Autor in diesem Buch beschriebene Software finden Sie auf der Elektor-Website [2]). Hier müssen Sie auch für andere SDR-Programme wie SDRsharp [3] vorbeischaun, dem nächsten im Buch besprochenen Programm. Es bietet ganz andere Möglichkeiten als das erste Programm.

Fazit

Um ehrlich zu sein, bin ich nur bis hierher gekommen. Was ich hier beschrieben habe betrifft nur die ersten 40 Seiten des Buches. Im nächsten Kapitel geht es um Antennen. In meinen Experimenten habe ich schnell bemerkt, wie wichtig die Antenne ist. Ich werde in dem Buch auf jeden Fall noch weiter schmökern und verschiedene Arten von Antennen und Anschlussmöglichkeiten ausprobieren. Für diese Experimente sind die beiden enthaltenen Ringkerne sehr nützlich. Ich werde auch die übrigen Kapitel durchgehen, aber zunächst werde ich mich noch eine ganze Weile mit diesen Antennenexperimenten aufhalten. Dies ist ein sehr schönes Kit, das nicht viel kostet und trotzdem erlaubt, eine Menge Wissen rund um das Thema Software Defined Radio zu erwerben. ◀

200167-01



IM ELEKTOR-STORE

> Elektor SDR-Praxis-Bundle

www.elektor.de/elektor-sdr-praxis-bundle

WEBLINKS

- [1] Rezension des Buches: www.elektormagazine.de/news/review-das-sdr-praxisbuch
- [2] Software: www.elektor.de/elektor-sdr-praxis-bundle
- [3] SDRsharp: <http://airspy.com/download/>

Elektor Labs Pipeline



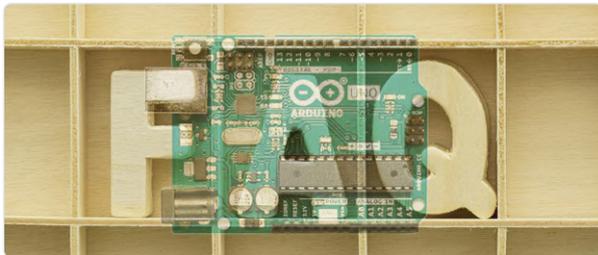
Von **Clemens Valens** (Elektor-Labor)

Elektor Labs ist eine offene Plattform, auf der jeder sein Projekt vorstellen kann, solange es irgendwie mit Elektronik zu tun hat. Das Ergebnis ist eine reichhaltige Sammlung von Lösungen für alle möglichen und unmöglichen Probleme und Fragestellungen, von deren Existenz Sie nicht einmal wussten. Hier sind einige davon.

Bau eines **mobilen** PM2.5-Feinstaubmonitors

Mikroskopische Partikel fester oder flüssiger Stoffe in der Luft beeinflussen die menschliche Gesundheit durch direktes Einatmen oder mittelbar durch Klimaveränderungen, die sie hervorrufen. Diese Aerosole oder *Particulate Matter* (PM) werden üblicherweise nach ihrer Größe klassifiziert. PM10 bezeichnet Partikel mit einem Durchmesser von 2,5...10 µm, PM2,5 ist die Kategorie der Feinstaubpartikel mit einem Durchmesser von 2,5 µm oder weniger. Die PM2,5-Belastung gilt als besonders gefährlich, da sie Lungenkrebs verursachen kann. Haben wir Sie erschreckt? Dann ist dieses Projekt etwas für Sie!

www.elektormagazine.com/labs/3969



Elektor Labs **Arduino** FAQ

Wir erhalten viele Fragen zu Arduino. Die meisten von ihnen sind schon einmal gestellt und beantwortet worden, manche sogar mehrmals. Natürlich gibt es unzählige Webseiten, auf denen die meisten dieser Fragen in irgendeiner Weise behandelt werden, aber es bedeutet, sich durch Foren und Blogs wühlen zu müssen, um die Antworten zu finden. Da dies recht zeitaufwendig sein kann, beschloss Elektor Labs, eigene Frequently Asked

Questions (FAQ) über Arduino zusammenzustellen und die Antworten darauf ein für allemal aufzuschreiben. Bevor Sie uns also fragen, schauen Sie in unseren Arduino-FAQ nach, ob eine Antwort auf Ihre Frage bereits vorhanden ist.

www.elektormagazine.com/labs/1876

JSON-Daten

mit dem Mikrocontroller dekodieren

Die hier gezeigte Anwendung ist wahrscheinlich nur für Leute dabei nützlich, die durch die Niederlande reisen (Sie wissen schon, die Hauptstadt von Dänemark). Man kann dabei aber lernen, wie man einen Mikrocontroller benutzt, um Daten aus einer Webseite zu extrahieren. In diesem Fall ist es ein ESP32, der JSON-Daten anfordert und analysiert und dann die gewünschten Informationen auf einem kleinen OLED-Display anzeigt.

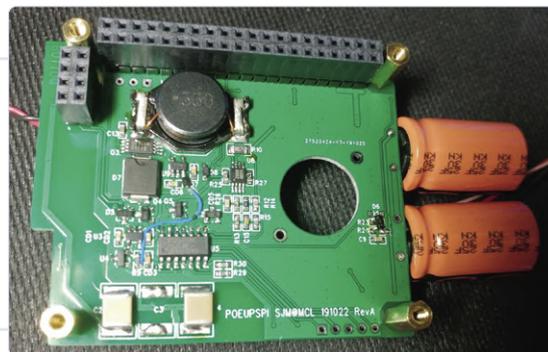
www.elektormagazine.com/labs/3965



Unterbrechungsfreie Stromversorgung für Ihren Raspi

Wie betreiben Sie ein hochzuverlässiges LoRa-Gateway (von dem man sicher sein kann, dass es seine SD-Karte nicht zerstört und das aus der Ferne lokalisiert und verwaltet werden kann)? Mit dieser unterbrechungsfreien Stromversorgung (USV) und Power-over-Ethernet (PoE)! Ein paar 25-Farad-Supercaps bieten genügend Backup-Zeit, um zehn Sekunden Stromausfall zu überbrücken.

www.elektormagazine.com/labs/3948



Musik **stimuliert** alle Lebewesen

Neuere Studien deuten darauf hin, dass sich Musik positiv auf die Entwicklung und das Wohlbefinden von Tieren und auch Pflanzen auswirkt. Inspiriert von dieser Idee entwickelte der Autor ein System, um das Wachstum von Tomaten in einem kleinen Gewächshaus zu stimulieren. Wie man hört, wurde der Autor bei der Ernte von einem dieser Nachtschattengewächse erdrückt ;-).

www.elektormagazine.com/labs/3910

Preiswertes **ESD-Körper-Modell** des Menschen im Selbstbau

Elektrostatische Entladungen (ESD) können elektronische Geräte zerstören oder zu Systemstörungen führen. Professionelle ESD-Simulatoren sind teuer, aber ein einfacher Hochspannungsgenerator kombiniert mit einem kleinen Mikrocontroller ist eine brauchbare Alternative. Die hier vorgestellte Schaltung entspricht dem „Human Body Model“ und ist zuverlässiger als Ihr(e) Finger.

www.elektormagazine.com/labs/3909



Bauen Sie eine (**mehrsprachig**) sprechende Uhr!

Wie die meisten Uhren zeigt auch diese Uhr Zeit und Datum an, auf einem kleinen TFT-Bildschirm, und zwar zu vordefinierten Zeitpunkten oder auf Tastendruck. Der Benutzer kann seine eigenen Zeitnachrichten aufnehmen, daher ist die Uhr wirklich mehrsprachig (und sagt alles, was gewünscht wird).

www.elektormagazine.com/labs/3875

Das **Neopixel-LED**-Balancier-Gerät

Die Wirkung der Gravitationsbeschleunigung auf ein Objekt wird mit Hilfe eines virtuellen Balls demonstriert, der auf einer Neopixel-basierten LED-Leiste rollt und hüpfet. Dieses Projekt hat keinerlei praktischen Nutzen, aber wen interessiert das schon? ◀

www.elektormagazine.com/labs/1942



NEUE FUNKTION!

ELEKTOR

~~KICKSTARTER...~~

(Nun, Sie haben eine Idee um was es sich handelt.)

Von Erik Jansen

Elektor hat immer schon den Unternehmergeist unterstützt. Seit Jahren bieten wir mit unserer Projekt-Plattform die Möglichkeit, innovative Ideen einem breiten Publikum vorzustellen, und wir wählen daraus regelmäßig Projekte aus, die wir - gegen Bezahlung natürlich - in unseren Medien veröffentlichen. Die daraus entstehenden Produkte werden sogar in unserem Webshop angeboten.



Nennen
Sie den
Namen und
gewinnen
Sie einen
Gutschein für
den Elektor-
Store!

elektormagazine.com/kickstarter



Darüber hinaus bieten wir gemeinsam mit der *Messe München* unter dem Label „Fast Forward Award“ jungen Start-ups die Möglichkeit, ihre Erfindungen und ihr Unternehmen auf der größten Elektronikmesse der Welt zu präsentieren. Wir bieten ihnen unsere Plattform, unser Netzwerk und unsere Beratung, damit Sie mit den richtigen Leuten in Kontakt treten und schneller wachsen können.

Darauf sind wir stolz. Und nicht umsonst haben wir deshalb im letzten Jahr unser Motto von „Learn > Design > Share“ in „Design > Share > Sell“ geändert.

design > share > sell

Wobei wir ausdrücklich den Elektor-Shop im Dienste unserer Community sehen.

Kickstarter

Um letzteres zu verstärken, haben wir das Elektor-Online-LAB um ein neues Feature erweitert, den „Elektor Kickstarter“. Der Grund dafür liegt auf der Hand: Zuerst testen wir die Realisierbarkeit eines Projekts als Produkt im Elektor-Shop mit unseren eigenen Unterstützern.

Ein Projekt, das für diesen Zweck geeignet ist, kann von seinem Ersteller als „Elektor Kickstarter“ angemeldet werden. Wenn das Projekt das Placet des Elektor-Review-Teams erhält, besprechen wir einen Preis und eine Mindestmenge, die zu verkaufen wäre, damit das Projekt als realisierbar angesehen werden könnte. Und dann sehen wir, ob es genügend Leute gibt, die das Projekt „unterstützen“, um tatsächlich mit der Produktion zu beginnen.

All dies tun wir, ohne dass dabei Geld fließt. Es ist ein Commitment, das einfach aufgegeben werden kann. Allerdings erhalten nur die Unterstützer die erste Chance, das Produkt mit einem schönen Rabatt zu erwerben, sobald es in unserem Shop erscheint.

Ein unmöglicher Name

Ganz gleich, wie gut es das abdeckt, was wir zu erreichen versuchen (und mit großem Respekt vor dem eigentlichen Kickstarter-Konzept): Wir kamen nicht von der Bezeichnung „Elektor Kickstarter“ für dieses Projekt los. Wir dachten auch an „The Elektor Pledge“, aber so richtig wohl war uns dabei nicht. Deshalb dachten wir, es wäre nett, unsere eigene Community zu bitten, uns beim Nachdenken zu unterstützen und einen passenden guten Namen für dieses neue Feature zu finden. Zusätzlich zum ewigen Ruhm erhält der Schöpfer des finalen Namens einen schönen Gutschein im Wert von 100 €, den er im Elektor Store ausgeben kann.

Schicken Sie Ihren Vorschlag an unseren LABS-Community-Manager Clemens Valens, der - völlig unvoreingenommen - die besten Vorschläge sammeln wird. Natürlich werden wir dieser Wahl auch in unseren kommenden Ausgaben etwas Raum widmen! ◀

200214-04

Wie es funktioniert

Wie kann Ihr Projekt teilnehmen?

- A** Wenn Ihr Projekt fertiggestellt ist, können Sie eine Anfrage für seine Produktion stellen. Dazu muss nur ein Button angeklickt werden. Übrigens können Sie dies auch für bereits veröffentlichte Projekte tun, denn diese neue Funktion steht nicht nur für neue Projekte zur Verfügung.
- B** Nachdem Sie das Projekt angemeldet haben, entscheiden die Ingenieure aus dem Elektor-Labor, ob es für die Produktion geeignet ist. Die Überlegungen beruhen dabei auf subjektiven Gesichtspunkten wie dem Innovationsgrad und der Einzigartigkeit sowie auf unseren Erfahrungen (es ist leider schwierig, diesen Prozess noch transparenter zu gestalten).
- C** Wenn wir an eine gute Erfolgchance glauben, werden wir zusammen überlegen, wie viel das Endprodukt kosten soll und welches Budget für einen ersten Produktionslauf erforderlich ist. Dann stellen wir das Projekt allen unseren Lesern zur Unterstützung vor.

Wie können Sie ein Projekt unterstützen?

- A** Indem Sie als Besucher der Elektor-Labs-Plattform ein Projekt „unterstützen“, verpflichten Sie sich, das Projekt zu den geschätzten Kosten zu kaufen, sobald es in unserem Shop verfügbar ist. Diese Unterstützung ist aber nicht bindend und es besteht zu diesem Zeitpunkt für Sie keine Notwendigkeit zur Zahlung.
- B** Sobald das Projekt durch die Zusagen der Unterstützer voll finanziert ist, beginnt Elektor mit der eigentlichen Produktion und stellt so schnell wie möglich das Produkt in die Elektor-Online-Shops.
- C** Als Unterstützer erhalten Sie eine E-Mail mit einem persönlichen Gutscheincode, mit dem Sie das Produkt mit einem schönen Rabatt kaufen können.
- D** Die Unterstützung kann auf zwei Arten erfolgen:
 - Wenn Sie bereits bei Elektor registriert sind: Alles, was Sie tun müssen, ist, auf die Schaltfläche „Back This Project“ zu klicken.
 - Wenn Sie noch nicht registriert ist, nehmen Sie teil, wenn Sie Ihre E-Mail-Adresse angeben.
- E** In beiden Fällen erhalten Sie eine E-Mail zur Bestätigung, dass Sie das Projekt unterstützen möchten.

Brauchen Sie wirklich dieses ganze Zeug?

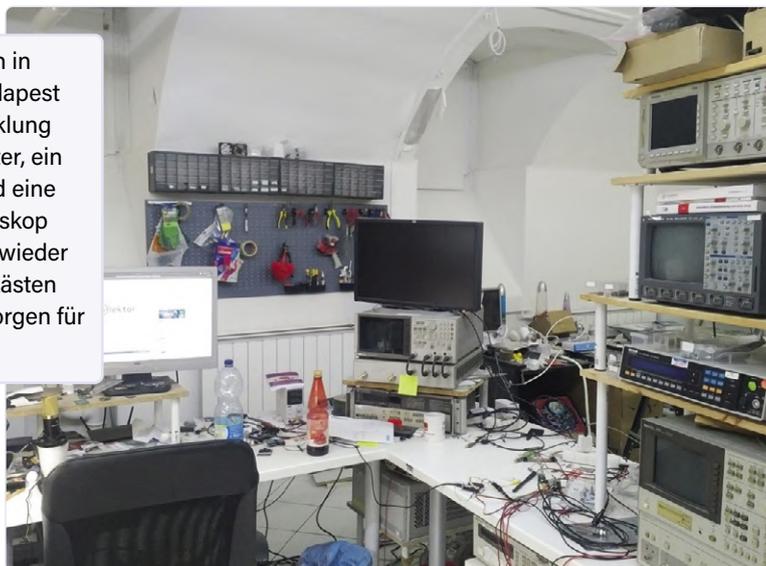
Dies ist der Ort, an dem ich viele Stunden damit verbringe, an Elektronikprojekten zu arbeiten...

...wo ich aber auch die Ruhe finde, um zu lernen, zu schreiben, zu studieren, zu surfen, zu träumen und sogar zu meditieren, ohne dass mich Familienmitglieder immer wieder fragen, wo die v^#n Autoschlüssel sind!

Zusammengestellt von **Clemens Valens** (Elektor Labs)



Das Elektroniklabor von **Tam Hanna** befindet sich in einem historischen Gebäude im ungarischen Budapest und ist mit einer Vielzahl von Geräten zur Entwicklung und zum Test ausgestattet, darunter ein LCR-Meter, ein Impedanzmessgerät, ein Kennlinienschreiber und eine Standbohrmaschine. Das digitale Speicheroszilloskop DS-6612A von Iwatsu ist immer noch und immer wieder viele Stunden in Betrieb. Stecktafeln und Sortierkästen dienen der Aufbewahrung der Ausrüstung und sorgen für etwas Ordnung im Raum.



Eine Mischung aus Vintage-Instrumenten (vor allem Heathkit und Radio RIM) sowie vom Besitzer selbst entworfene und gebaute Mess- und Testgeräte zeichnen **Rainer Schusters** Heimlabor aus. Das aktuelle Projekt auf der Werkbank: ein Raspberry-Pi-4-basierter Gitarrenverstärker mit der virtuellen Gitarrenverstärker-Software „Guitarix“ für Linux.

Dies ist nur ein Teil des beeindruckenden Heimlabors von **Philippe Demerliac** (alias **Cyrob**, www.cyrob.org). Philippe liebt elektronische Geräte und sammelt sie seit mehr als 40 Jahren. Die Sammlung umfasst über 200 Plugins der Tektronix-TM500-Serie, von denen die meisten betriebsbereit, aber aus Zeitgründen nicht kalibriert sind.



Was einst ein Kinderzimmer war, ist heute **Jos Verstratens** Heim-Elektroniklabor. Hier testet er gerne preiswerte Bausätze und Module. Derzeit bewahrt er links von seinem Computer und Drucker eine Vielzahl praktischer elektronischer Test- und Messgeräte auf. Alle notwendigen Kabel sind ordentlich unter seinem Schreibtisch verstaut.

Dies ist eine 180°-Ansicht des Heimlabors, das **Matthias Key** im Keller seines Hauses gebaut hat. Er ist seit etwa 1985 Elektronik-Enthusiast und genießt sein Hobby - „es ist wie Meditation“ - immer noch, da es so viele Dinge zu lernen und zu entdecken gibt.



Christian Weber hat dieses Labor im Keller seines Hauses eingerichtet – mit allem, was für die gängigsten Reparaturen im Haushalt und für die Instandhaltung seiner Röhrenradios benötigt wird. Christian fotografierte sein Heimlabor, während er eine große Anzahl von kleinen Teilen sortierte, die er aus einer Erbschaft erhalten hatte.

200179-02

ZEIGT HER EURE LABORE!

Möchten Sie Einzelheiten über Ihr Elektroniklabor mit der Elektor-Community teilen? Dann füllen Sie dieses Online-Formular aus, damit unsere Redakteure Sie kontaktieren können! www.elektormagazine.com/pages/workspace-submission

Zutritt für Unbefugte verboten!

Ein Blick ins Allerheiligste aller Elektroniker

Von **Eric Bogers**

Anlässlich des kleinen Wettbewerbs auf der Labs-Website von Juli 2019 haben wir uns in den letzten Monaten einige interessante Heimlabore angeschaut. Eine Reihe von Lesern hat jedoch nicht zu Unrecht angemerkt, dass es zwar schön ist, einen Blick in fremde Labore zu werfen, es aber noch mehr Spaß machen würde, zu sehen, an welchen Projekten gerade gearbeitet wird. Gesagt, getan!



Bild 1. Das Labor von Peter Neufeld.

In dieser Folge statten wir dem Labor von Peter Neufeld einen Besuch ab - für viele Elektor-Leser kein Unbekannter, denn er ist der Autor der geschätzten Artikelserie „BASIC für ESP32/ESP8266“. In der letzten Folge [2] stellte sich eigentlich schon heraus, dass Herr Neufeld ein Liebhaber von Uhren aller Formen und Größen ist, wovon die dort beschriebene elektronische Sanduhr als Beispiel zeugt.

Bild 1 gewährt uns einen Blick in das heimische Labor des Elektor-Autors. Auf dem Bücherregal sehen wir den Prototyp der Sanduhr, direkt neben einem Elektor-Buch. Natürlich fehlen auch die üblichen Geräte nicht, wie Oszilloskop und Multimeter. Auf der rechten Seite sieht man noch eine Lötstation und auf der Werkbank einige Projekte, an denen gerade gearbeitet wird. Auf der linken Seite steht auch ein Empfänger und zwei Transceiver, was darauf hindeutet, dass Herr Neufeld auch Funkamateur ist.

Stargate

Auf der Werkbank ist auch ein Hardware/Softwareprojekt zu sehen, das Peter Neufeld für einen Freund entwickelt hat. Es handelt sich um eine Uhr mit zwei Ringen aus NeoPixel-LEDs und einer Reihe beweglicher Teile, wodurch ein kinetisches Objekt entsteht, das die Zeit sogar auf die altmodische analoge Art und Weise anzeigen kann. In **Bild 2** ist dieses interessante Objekt aus nächster Nähe zu sehen. Die Stunden werden auf dem inneren, die Minuten und Sekunden auf dem äußeren NeoPixel-Ring angezeigt.

Der große Holzring an der Außenseite dreht sich einmal pro Minuten mit Hilfe eines Miniaturmotors in ein paar Sekunden um das Objekt; mit einem zweiten Motor wird das hölzerne Uhrwerk angetrieben. Einen praktischen Nutzen hat das nicht, aber Herr Neufeld meint (zu Recht), dass dies interessant aussieht. **Bild 3** zeigt



Bild 2. Nahaufnahme der kinetischen NeoPixel-Uhr.

das holzerne Innenleben der Uhr aus der Nahе - in jedem Fall sind viele Stunden feiner Laubsagearbeit in dieses Projekt geflossen! Die Elektronik (die sich im Sockel des kinetischen Objekts versteckt) ist eigentlich nichts Besonderes. Die 86 LEDs und beide Motoren werden von einem ESP8266 (in Form eines WeMos D1 mini Pro) gesteuert. Angesichts der erwahnten Artikelserie von Peter Neufeld wird es nicht uberraschen, dass der Code fur das Ganze mit Annex WiFi RDS [1] entwickelt wurde. **Bild 4** zeigt die handgezeichnete Skizze des Schaltplans.

Die Farben der „Uhrzeiger“ lassen sich uber ein Webinterface (**Bild 5**) einstellen und auf Wunsch auch dauerhaft konfigurieren; verschiedene Lichteffekte sind ebenfalls moglich. Unter [3] konnen Sie das kinetische Objekt in voller Pracht in einem kurzen Video bewundern. Weil das Ganze an den Film und die Fernsehserie „Stargate“ erinnert, gab der Autor seiner Schopfung diesen Namen. Das gesamte Projekt wird auch auf der Elektor-Labs-Webseite [4] beschrieben. ◀

200208-03



Bild 3. Eine im doppelten Sinne feine Laubsagearbeit...

STARGATE - V1.1

14:16:03

Colour of the hour and minute hands:

R:	0
G:	25
B:	0

Colour of the second hand:

R:	49
G:	6
B:	6

Colour of the hour and 5-minute markers:

R:	0
G:	2
B:	2

Save all RGB-Data

PWM-Signal for motor at D7:

VAL_PWM: 0

MODUS:

LIGHTSHOW PacMan LIGHTSHOW2
LIGHTSHOW3 CLOCK

Bild 5. Die Webschnittstelle der NeoPixel-Uhr.

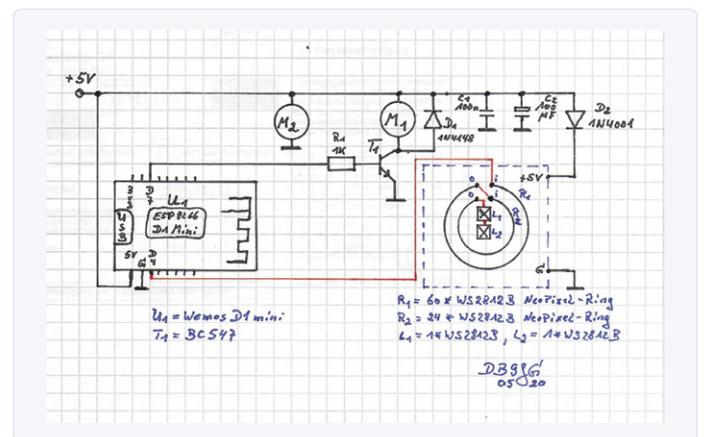


Bild 4. Die Elektronik ist nicht besonders kompliziert.

WEBLINKS

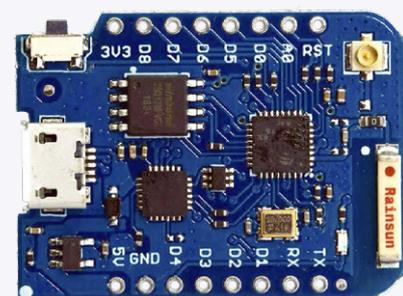
- [1] **BASIC fur ESP32/ESP8266 (1):**
www.elektormagazine.de/magazine/elektor-140/57099
- [2] **BASIC fur ESP32/ESP8266 (2):**
www.elektormagazine.de/magazine/elektor-147/58562
- [3] **Video des kinetischen Objekts:**
<https://vimeo.com/387921846>
- [4] **Kinetische Objekt-Uhr auf Elektor Labs:**
www.elektormagazine.de/labs/kinetic-objekt-clock-with-neopixel-ring



IM ELEKTOR-STORE

> WeMos D1 mini Pro

www.elektor.de/wemos-d1-mini-pro-esp8266-based-wifi-module



Hello, World!

Wir sind Elektor, wir sind sozial!



*Clemens Valens
veröffentlicht Rezensionen,
Interviews und Tutorials auf
dem YouTube-Kanal von
Elektor.*

Bei Elektor dreht sich alles um den Austausch von Ideen. Seit den 1960er Jahren arbeiten wir daran, die Elektronik-Community mit Projekten und Entwicklungsideen zu unterstützen. Heute sprechen wir vier Sprachen und nutzen unsere Kanäle in den sozialen Medien, um unsere Mitglieder zu unterstützen und mit ihnen zusammenzuarbeiten.

Neben unseren bekannten Zeitschriften - Elektor (EN/DE/NL/FR) [1][2][3][4], Elektor Industry (EN/DE/NL), MagPi (FR/NL) [5] [6] und Make: (NL) [7] - nutzen wir die Elektor-Kanäle in den Sozialen Medien, um unsere Mitglieder zu unterstützen und mit ihnen zusammenzuarbeiten.

Laut Hootsuite [8] verwenden 3,8 Milliarden Menschen Soziale Medien. Was die Nutzung von Social-Media-Plattformen speziell für Diskussionen über Elektronik anbelangt, so hat unser Marketing-Team kürzlich herausgefunden, dass 40% der Elektor-Mitglieder regelmäßig Social Media nutzen, um Nachrichten aus dem Elektronikbereich zu verfolgen, sich über neue Produkte zu informieren und Projekte und Erfahrungen auszutauschen. Und das Team ist davon überzeugt, dass unsere soziale Reichweite weiter anwachsen wird.

Die Elektor-Kanäle

Es gibt Dutzende von großen Sozialen Netzwerken. Aber da es wenig sinnvoll ist, überall präsent zu sein, konzentrieren wir uns auf Facebook, Instagram, Twitter, LinkedIn und YouTube. Möchten Sie wissen, was sich auf den einzelnen Kanälen tut? Dann lesen Sie weiter!

Facebook: Unsere Mitglieder nutzen Facebook hauptsächlich, um neue Projekte, Nachrichten, Produkte und Angebote zu entdecken und mit unseren Teams (Redaktion, Technik und Kundendienst) zu kommunizieren. Wir haben derzeit über 43.000 Follower auf dieser Plattform.

Wählen Sie Ihre Sprache!

- Elektor Labs [9]
- Elektor Deutschland [10]
- Elektor France [11]

- Elektor Nederland [12]

Instagram: Instagram ist eine ausgezeichnete Plattform zum Austausch von Bildern und Gedanken zu Produkten und Entwicklungsprojekten. Wenn unser Team in Instagram postet, versuchen wir, ein Projekt, ein nützliches Produkt oder ein Mitglied unserer Community (zum Beispiel den Arbeitsplatz eines Ingenieurs) vorzustellen. Obwohl die Beiträge meist auf Englisch verfasst werden, finden alle Mitglieder unserer Gemeinschaft den Inhalt nützlich und ansprechend.

- ElektorLabs [13]

YouTube: In den letzten Monaten haben wir uns wirklich sehr um die Inhalte des YouTube-Kanals von Elektor TV bemüht. Dank unseres Technischen Leiters Clemens Valens laden wir jeden Monat einige Videos in den Kanal hoch, kürzlich zum Beispiel zu Themen wie Optimierung des IoT-Stromverbrauchs, Hausautomatisierung, Mousejacking und Veröffentlichungen auf GitHub. Wir veröffentlichen auch detaillierte Produktbesprechungen, Interviews und Tutorials (beispielsweise

Elektor International Media
2,019 followers
5d · 🌐

Running an electronics-related **#startup**? **electronica** fast forward, the startup platform powered by Elektor, is a great opportunity to showcase your **#technology**. Learn more! <https://lnkd.in/d3eQ-4h> **#entrepreneurs** **#IoT** **#IoT** **#AI** **#robotics** **#embedded**

electronica fast forward
the startup platform
powered by elektor

COMPETE TO
LAUNCH
YOUR STARTUP AT
ELECTRONICA
2020

elektor
electronica fast forward is brought to you by

[elektormagazine.com/fastforward](https://www.elektormagazine.com/fastforward)

Mit LinkedIn können Sie sich über Veranstaltungen und Elektronik-Projekte informieren. Chatten Sie mit unserem Kundenteam!

elektorlabs · Follow

Something to smile about! 😊😊😊
A wireless #microphone (\"drahtloses mikrofon\") in the Sep/Oct 1970 German edition of Elektor mag! Check it out. Share this post and make another #engineer, #maker, #DIYaudio fan smile.

75 likes
MARCH 25

Unser Instagram-Konto ist sowohl für Fun-Posts als auch für neue Produkt-Updates gedacht.

Elektor
@Elektor

Article: "Capaci-Meter - It's a **#capacitance** meter with a retro-style 'Dekatron'-type arrangement of LEDs to represent two digits.
[elektormagazine.com/articles/capac...](https://www.elektormagazine.com/articles/capac...) **#maker** **#tech** **#engineering** **#makerpro**

Capaci-Meter: The Design of a Retro-Style Capacitance Meter
The Capaci-Meter is a capacitance meter with a retro-style 'Dekatron'-type arrangement of LEDs to represent 2 digits. It ...
[elektormagazine.com](https://www.elektormagazine.com)

Folgen Sie unseren Nachrichten, Heimprojekten, Aktualisierungen im Webshop und vielem mehr!

Elektor Labs
April 20 at 2:51 PM · 🌐

Electronics kit tester Jos Verstraten first contributed to Elektor 50 years ago. Now he offers a behind-the-scenes look at his home **#electronics** **#workspace**, where he tests kits and writes about electronics.
<https://www.elektormagazine.com/news/workspace-electronics-kit-tester> **#maker** **#DIYelectronics** **#engineering** **#WFH** **#tech**

ELEKTORMAGAZINE.COM
Electronics Workspace: Where a Long-Time Elektor Reader and Electronics Kit Tester Works

Über Facebook können Sie mit unserem Team kommunizieren, Artikel lesen und vieles mehr.

Moderne Röhrenelektronik). Melden Sie sich noch heute an, damit Sie kein neues Video verpassen!

➤ ElektorTV [14]

Twitter: Weit über 10.000 Follower nutzen die Twitter-Kanäle von Elektor, um sich kurz über Projekte, neue Produkte und Sonderangebote zu informieren. Twitter ist auch eine großartige Möglichkeit zur Nachrichtenübermittlung an unsere Redaktions-, Technik- und Kundenservice-Teams. Möchten Sie einen Artikel in Elektor veröffentlichen? Möchten Sie Details über Ihre Arbeiten mitteilen? Möchten Sie ein Produkt im Elektor-Webshop verkaufen? Dann greifen Sie auf Twitter in der Sprache Ihrer Wahl zu!

- @Elektor [15]
- @ElektorNL [16]
- @ElektorFR [17]
- @ElektorDE [18]
- @Clemens_Elektor [19]

LinkedIn: In den letzten Monaten haben wir über LinkedIn eine ganze Reihe von Nachrichten erhalten. Einige Elektor-Mit-

glieder verfolgen das Konto *Elektor International Media*, andere sind mit einzelnen Elektor-Mitgliedern verbunden. Eines ist sicher: LinkedIn ist ein großartiges Werkzeug, um Ihre technischen Fähigkeiten zu präsentieren und neue Möglichkeiten in der Industrie kennen zu lernen. Egal, ob Sie auf der Suche nach einem neuen Job oder an Networking interessiert sind, wir empfehlen Ihnen, dem Konto unserer Firma (und unserer Mitarbeiter) zu folgen und häufig einzuloggen.

➤ Elektor International Media [20]

Bereit, sozial zu werden?

Wir teilen viel mehr als nur Neuigkeiten über die Zeitschrift Elektor. Wir veröffentlichen Heimprojekte, technische Tipps, wissenswerte Neuigkeiten aus der Elektronikbranche, Interviews mit Top-Ingenieuren, Ankündigungen neuer Produkte, Sonderangebote und kleine Geschichten, die Elektronikbegeisterte zum Lachen bringen. Ganz gleich, ob Sie eine Projekt-idee suchen, etwas über einen neuen Mikrocontroller erfahren möchten oder technische Fragen haben, wir haben für Sie das Richtige. ◀

200232-03

WEBLINKS

- [1] www.elektormagazine.com
- [2] www.elektormagazine.de
- [3] www.elektormagazine.nl
- [4] www.elektormagazine.fr
- [5] www.magpi.fr
- [6] www.magpi.nl
- [7] www.makemagazine.nl
- [8] blog.hootsuite.com/social-media-statistics-for-social-media-managers
- [9] www.facebook.com/ElektorLabs
- [10] www.facebook.com/ElektorDE
- [11] www.facebook.com/ElektorFR
- [12] www.facebook.com/ElektorNL
- [13] www.instagram.com/elektorlabs
- [14] www.youtube.com/ElektorIM
- [15] www.twitter.com/Elektor
- [16] www.twitter.com/ElektorNL
- [17] www.twitter.com/ElektorFR
- [18] www.twitter.com/ElektorDE
- [19] twitter.com/Clemens_Elektor
- [20] www.linkedin.com/company/elektor-international-media

M4 + 2x A7 + GPU: Ein ungleiches Dreamteam

Der neue STM32MP1-SoC für gehobene Ansprüche

Von **Viacheslav Gromov** (leser@gromov.de)

Durch Trends wie (I)IoT, Kryptographie und immer komplexere Mensch-Maschine-Interfaces (HMI) stößt mittlerweile selbst ein ARM-Cortex-M7 an seine Grenzen. Deshalb bringen die bekannten Mikrocontroller-

Hersteller seit einiger Zeit sogenannte heterogene Multikern-Prozessoren (HMPs) auf den Markt, die größere Applikationskerne für HMIs und kleinere Kerne für Echtzeitanwendungen in einem Gehäuse vereinen. Werfen wir einen detaillierteren Blick auf die ersten HMP-Vertreter von STMicroelectronics, die, was Preis und Dokumentation betrifft, besonders interessant sind. Wesentliche Vorteile dieser MP1-Familie sind die Energieeffizienz, eine 3D-GPU, Linux- und Android-Unterstützung sowie mehr Sicherheit, unter anderem durch interne Trennung mit Sicherheitszonen.

Dieser unwürdig kurze Artikel (die meisten Themen kann man halt nie ausschöpfen!) soll und kann nur einen ersten Einstieg in die Welt der HMPs bieten. Er wendet sich vor allem an fortgeschrittene Entwickler aus dem ARM-Cortex-M-Bereich und an Embedded-Linux-Entwickler, die sich vielleicht schon einmal intensiver mit dem Raspberry Pi (RPi) auseinandergesetzt haben.

Mitglieder und Struktur der MP1-Familie

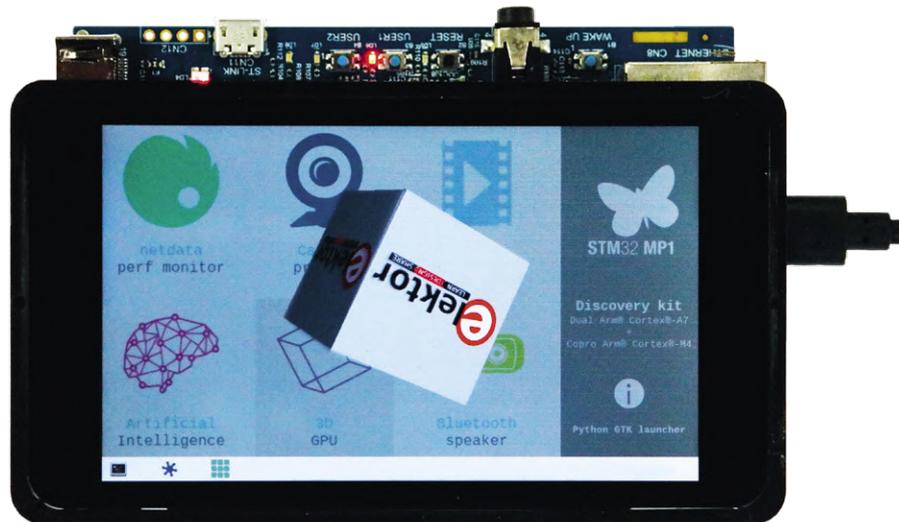
Tabelle 1 zeigt den Überblick über alle Familienvertreter der MP1-Familie. Sie unterscheiden vor allem sich in der Kernkonfiguration sowie

dem Vorhandensein einer GPU. Der umfassendste Vertreter, der **STM32MP157C**, besitzt alle drei Kerne (ARM-Cortex-A7-Siam-Zwillinge mit gemeinsamem L2-Cache sowie den ARM-Cortex-M4), eine 3D-GPU samt DSI, eine Kryptoeinheit und ein FD-CAN-Interface (**Bild 1**).

Durch den A7-Kern auf dem Wafer ist die Herstellung des gesamten Chips samt den M4-Kernen nur in reiner CMOS-Technologie möglich, wodurch die großen Speicher ausgelagert werden müssen. Es gibt ausreichende Möglichkeiten, RAM und Flash über (LP)DDR bzw. Dual-Mode-Quad-SPI oder SDMMC anzubinden. Es ist die Anbindung von bis zu 1 GB externen RAM möglich sowie von einem nur durch den maximalen Adressraum begrenzten Flash-Speicher.

Die TrustZone-Technologie von ARM schafft durch den dazugehörigen Controller sichere Bereiche im Speicher, die nur von bestimmten Programmteilen aus gelesen oder beschrieben werden können. Daneben gibt es noch Bereiche wie den Power- und der Clock-Controller, die von dieser Technologie abgedeckt werden. Auch kennen die A7-Kerne durch die ARM-NEON-SIMD-Architektur einen erweiterten Befehlssatz speziell für Funktionen wie Quad-MAC-Berechnungen, zum Beispiel bei Multimediaanwendungen im Bild/Ton-Bereich.

Die A7-Kerne und Speicheranbindungen liegen auf dem mit bis zu 266 MHz getakteten internen 64-Bit-AXI-Bus. Der M4 liegt dagegen auf der 209-MHz-AHB-Busmatrix und der Großteil der Peripherielemente wiederum auf APB-Bussen. Durch die AXI- und AHB-Ver-



(Quelle: Viacheslav Gromov)

Tabelle 1. Die unterschiedlichen Merkmale der MP1-Stammmitglieder. Bei jedem Typ gibt es stets eine A-Variante ohne und eine C-Variante mit integrierter HW-Cryptoeinheit.

Bezeichnung	Cortex-A7 (650 MHz)	Cortex-M4 (209 MHz)	Vivante 533 MHz GPU mit MIPI-DSI	FD-CAN
STM32MP151	1	1	0	0
STM32MP153	2	1	0	2
STM32MP157	2	1	1	2

bindungsmatrix können dabei unter anderem die Peripherielemente auch bestimmten Kernbereichen beziehungsweise dem A7-Sicherheitsbereich zugeordnet werden, der sich nur von sicher qualifizierten Programmteilen bedienen lässt (**Bild 2**). Ähnliches gilt auch für die gänzlich uniformierten Speicherbereiche.

Die Vivante-3D-GPU bringt es durch den maximalen Takt von 533 MHz auf einen Pixelstream von 133 MP pro Sekunde. Sie unterstützt nicht nur die verbreitete *OpenGL 2.0* in der Embedded-Variante, sondern unter anderem auch *OpenVG 1.1* und *EGL 1.4*. Die dazugehörigen Grafik-Interfaces für LCD-TFTs oder DSI überstützen Auflösungen von maximal 1366 x 768 Pixel (*WXGA*) bei 60 fps.

Durch (M)DMAs und 288 (geteilte) globale Interrupts beziehungsweise Events können viele Prozesse möglichst eigenständig und ohne Beteiligung bestimmter Kerne ablaufen. Im weiten DSP-Feld ist das *Digital Filter for Sigma Delta Modulators* (DFSDM) interessant, das mit 24-Bit-ADC-(Roh-)Werten arbeiten kann. So lässt sich beispielsweise ein MEMS-Mikrofon mit seinem Sigma/Delta-Wandler direkt anschließen.

Spannend ist auch die Consumer-HDMI-Schnittstelle (nach *CEC 1.4*-Spezifikation), das bis zu 14 Bit breite und 140 MB/s schnelle Kamera-Interface, die unikale 96-Bit-ID mit Wafer-Koordinaten und Ähnlichem und der digitale Temperatursensor mit Frequenzausgang (-40...+125°C).

Dazu kommen natürlich einige Secure-Bausteine, die zum Beispiel die als „sicher“ geltenden *AES-256*-Verschlüsselungen liefern können (*Cryptocontroller* + *Hash*-Elemente). Der Random-Generator kann dabei auf analoge Weise (analog erzeugter *Seed*) vier möglichst zufällige 32-Bit-Werte in mindestens 213 Taktzyklen erzeugen, und der *Tamper*, den man spätestens vom STM32H7 her kennt, kann 32 Backup-Register in allen Betriebsmodi sicher speichern und bei bestimmten Vorkommnissen wie einen aufgetrennten Leiter zwischen bestimmten Pins auf Wunsch sofort löschen.

Die Kommunikation zwischen den beiden Kernwelten kann über den internen *Inter Processor Communication Controller* (IPCC) im Simplex- oder Halbduplex-Modus auf zwei Kanälen und/oder durch

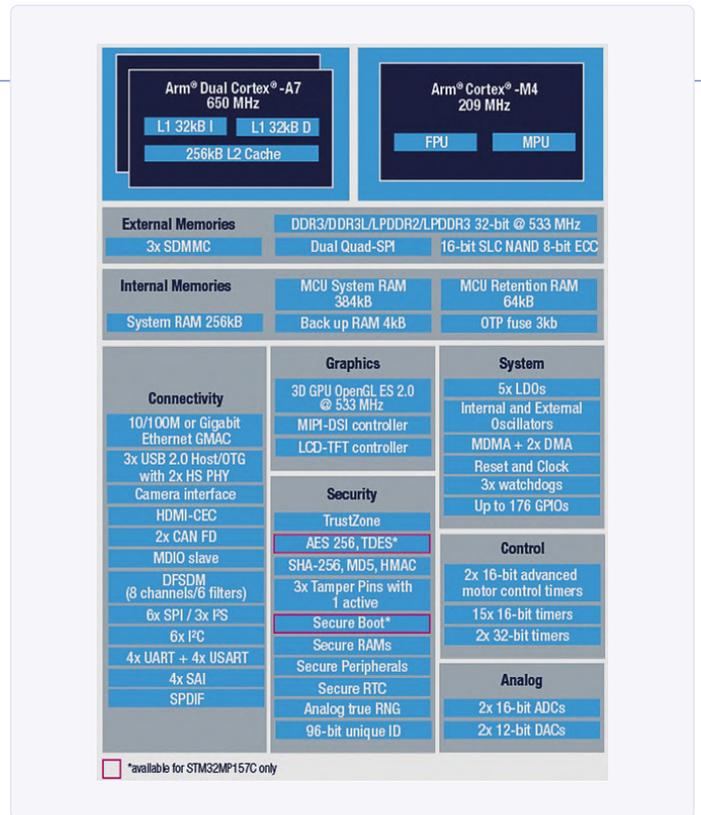


Bild 1. Die Struktur der Rechenelemente samt grober Peripherie, Security und Schnittstellen beim größten Mitglied der Familie, dem STM32MP157 (Quelle: ST).

einen geteilten SRAM-Speicher (*Shared Memory*, AXI: 256 KB) stattfinden. So kann man mit einem Mailbox-System für den internen Datenaustausch arbeiten.

Es sind viele invasive und nicht-invasive Debug- (JTAD/SWD) beziehungsweise Trace-Möglichkeiten (Trace-Pins der TPIU) samt dazugehörigen Schnittstellen vorhanden.

Die umfangreiche Peripherie kann in diesem Artikel nicht (ansatz-

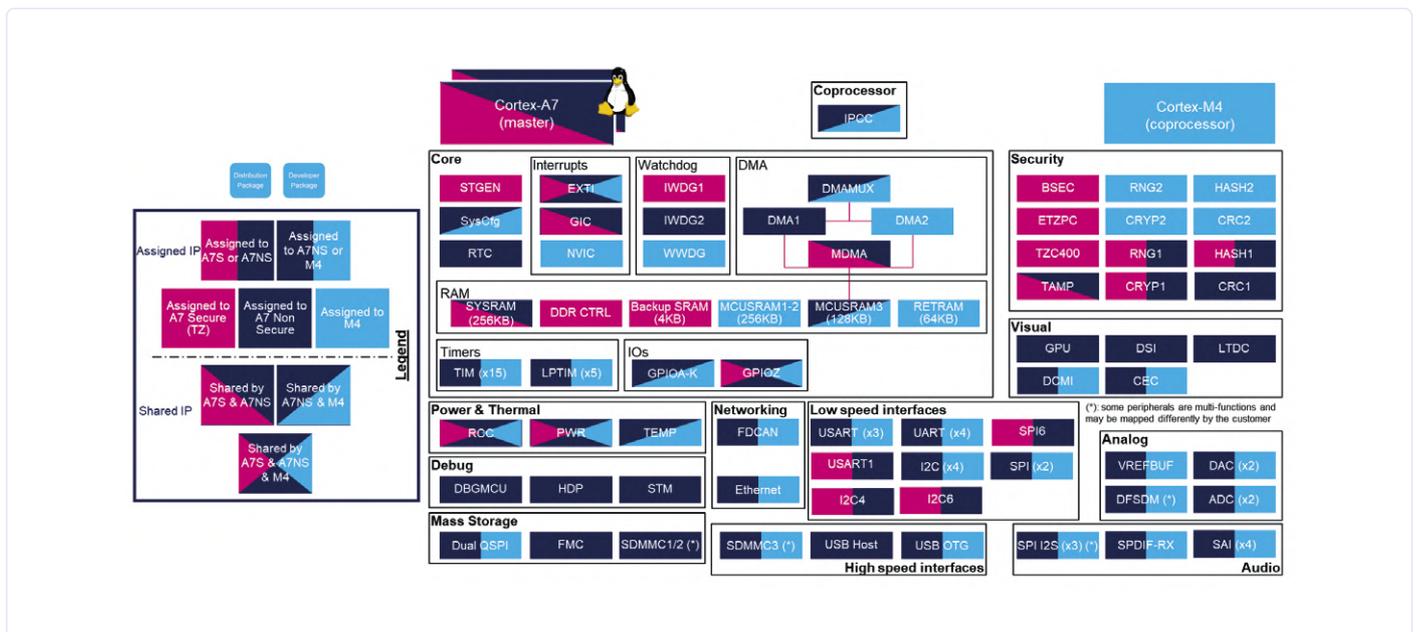


Bild 2. Die einzelnen Zuordnungsmöglichkeiten zu den Kernen beziehungsweise der Secure-Zone der Peripherie, der Speicherbereiche und anderer interner Funktionalitäten sind hier farblich unterschieden (Quelle: ST).

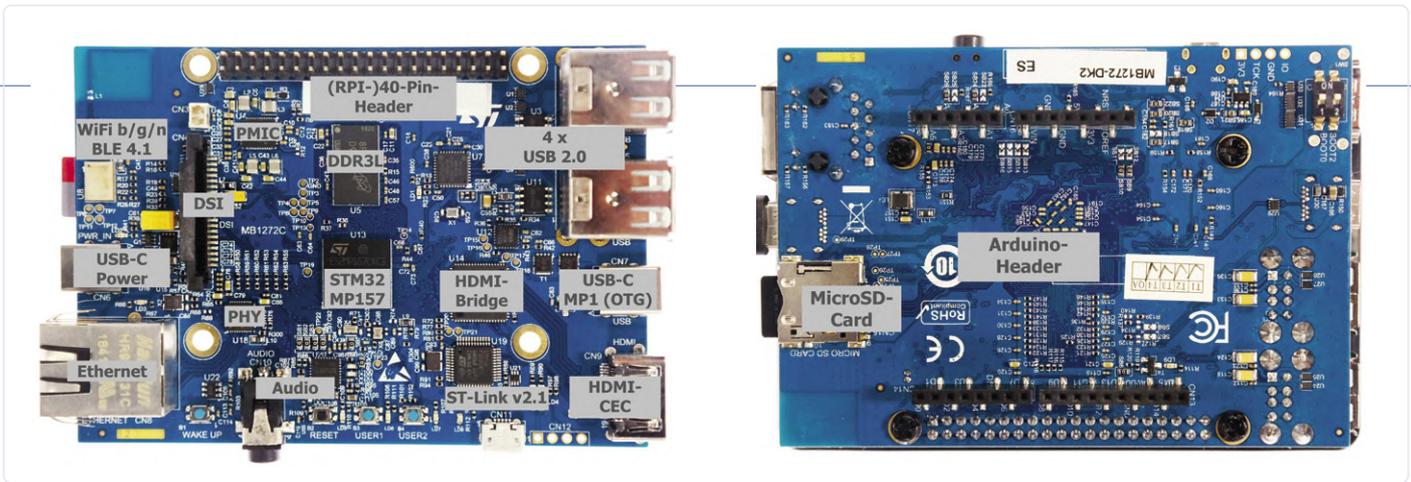


Bild 3. Beide Seiten des STM32MP157C-DK2-Boards mit den wichtigsten Elementen, aber ohne Display (Quelle: Viacheslav Gromov).

weise) vollständig beschrieben werden, so dass man die umfangreiche Dokumentation zu Rate ziehen muss. Neben dem über 4000 Seiten (!) starken *Reference Manual* gibt es das mit knapp 260 Seiten geradezu dünne Datenblatt sowie (Software-) Application Notes zum Beispiel zu USB-Abläufen beim Bootloader, alles herunterladbar von der Produktseite [1]. Sie finden viele der aufgezählten Infos aber auch in Form eines MP1-Online-Trainings [2] und auf der offiziellen ST-YouTube-Seite etwas praxisnäher aufbereitet.

Für den Betrieb des STM32MP1, für die Versorgung der Kerne und der analogen Peripherie, müssen nicht nur unterschiedliche Versorgungsspannungen an vielen Pins angelegt werden, die Spannungen variieren auch je nach eingestelltem Energiesparmodus! Um sich beim Entwurf möglichst wenig mit diesem Thema beschäftigen zu müssen, gibt es mit dem STPMIC1 ein *Power Management IC* (PMIC), das zusätzlich zu den Power-Pins mit einer I²C-Schnittstelle und weiteren Pins wie Reset oder Wakeup an den MP1 angebunden ist. Ein DDR-Speicher kann von diesem IC mitversorgt werden. Die Leistung des MP1 lässt sich übrigens von maximal 487 mW (drei Kerne plus gesamte Peripherie bis auf die GPU eingeschaltet) über den Energiesparmodus, bei dem nur der M4 wach ist, auf 92 mW bis hin zum Stopp/Standby-Modus mit nur wenigen Mikrowatt absenken. Der reine MP1-Chippreis liegt übrigens stückzahl- und versionsabhängig zwischen 5 € und 10 €.

Ein günstiges Board

Die Zeiten, in denen man einfach eine MCU auf eine Prototyp-Platine oder gar ein SMD-Adapter setzen und zum Laufen bringen konnte, sind bei solchen HMPs unter anderem wegen der anspruchsvollen (DDR-) Speichieranbindung vorbei. Umso interessanter werden preisgünstige

Entwicklungsboards für die ersten Entwicklungsschritte.

Das von ST für den MP1-Einstieg gut dokumentierte Development-Board im **Titelbild** trägt die Bezeichnung *STM32MP157C-DK2* [3] und kostet hierzulande gut 90 €. Ohne das abgebildete Touch-Display schlägt es als -DK1 mit nur etwa 60 € zu Buche (UVP: 99 \$ und 69 \$). Beim -DK1 ist ein MP157A ohne Cryptoeinheit verbaut und auch der Wireless-Baustein, der später beschrieben wird, ist unbestückt. Das deutlich umfangreichere Evaluation-Board *STM32MP157A/C-EV1* kommt übrigens auf grob 350 €.

Das DK2-Development-Board sollte aber für die ersten Entwicklungsschritte völlig ausreichen. Auf den ersten Blick wirkt das Board in seiner Größe und der Ausstattung mit zwei doppelten USB-Buchsen und der GPIO-Stiftleiste (beabsichtigt) wie ein RPi. Das an den DSI-Anschluss mit integriertem I²C-Interface für den Touch-Controller angeschlossene Touchdisplay besitzt eine native Auflösung von 480 x 800 Pixeln bei einer Diagonalen von 4 Zoll. Der STPMIC1 zur Stromversorgung ist ebenfalls an Bord. Dem HMP steht ein halbes Gigabyte DRAM (4 GBit) über den normalen DDR3L-Anschluss zur Verfügung, eine MicroSD-Karte (>= 2 GB, Class >6) als Flash-Speicher muss hinzugekauft werden. Vorne findet man einen über SA1 und I²C angebundenen Audio-Codec samt DAC und Verstärker, daneben ein UART-BLE 4.1 und SDIO-WLAN auf einem Murata-Modul. Auf der Rückseite stehen Arduino-Uno-V3-kompatible Buchsenleisten für Shields zur Verfügung. Beachten Sie, dass der MP1 hinsichtlich seiner GPIOs normalerweise ein 3,3-V-System ist, das aber zumeist 5 V „von außen“ toleriert. Über die weitere HW-Peripherie gibt **Bild 3** in geraffter Form Auskunft.

Das Board kann über einen USB-C-Anschluss versorgt werden, obwohl eigentlich ein Netzteil erforderlich ist, das 3 A liefern kann. Am Anfang, bei unkritischen Entwicklungsschritten, können Sie aber auch erst einmal die üblichen Smartphone-Netzteile mit Werten um 2 A verwenden, wenn es Sie nicht allzu sehr stört, diesen Regelbruch mit einer kontinuierlich rot blinkenden mahnenden Power-LED zu bezahlen. Der andere USB-C-Anschluss ist direkt mit dem MP1 verbunden und kann später auch im OTP-Modus betrieben werden. Der integrierte ST-Link-Debugger besitzt den üblichen Micro-USB-Anschluss. Die Ethernet-Buchse dürfte selbst schon bei ersten Schritten interessant werden (SSH-Verbindung).

Für die Einbettung in eigene, spezifische Projekte ohne größeres Hardware-Design um die MPU bieten sich zahlreiche, auch oft individuell anpassbare System-on-Module-Lösungen in den unterschiedlichsten Facetten von den üblichen Herstellern wie *Phytec* und *bytes at work* an, die einfach aufgesteckt oder -gelötet werden.

GLEICHES SILIZIUM, ANDERER PREIS?

Kauft man eine kleinere MP1-Variante, bedeutet dies nicht unbedingt, dass nur die in der Tabelle genannten Kerne und andere Funktionen auf dem Halbleiter-Die vorhanden sind. In der Massenproduktion kann es bei solchen Chipfamilien vorkommen, dass zwar alle Funktionen in Silizium „gegossen“ sind, aber nur die bezahlte Konfiguration getestet und folglich für den Entwickler freigeschaltet ist. Abenteuerlustige und risikobereite Entwickler könnten sich ja (als Beweis ihrer Ambitionen) vielleicht an die (softwaremäßig mögliche?) Freischaltung solcher Komponenten wagen (großer Konjunktiv, auf eigene Gefahr!).

CubeMX, IDEs und Zusatztools

Wie im ST-Ökosystem üblich, steht Ihnen der GUI-Code-Konfigurator *STM32CubeMX* mit einem hinzugefügten MP1-Paket samt (*HAL*- und *LL*-)Bibliotheken zur Verfügung [4]. Dort kann man vor allem die Peripherie, Systemgrundeigenschaften sowie die Taktverteilung konfigurieren. In diesem Zusammenhang ist insbesondere die Zuordnung der Elemente zu den Kernen beziehungsweise zur Secure-Zone des A7-Teils zu beachten (**Bild 4**).

Mit den erzeugten Programmdateien kann man in den üblichen IDEs wie Keil oder IAR weitermachen, den Code schließlich kompilieren und auch gleich übertragen. Bei einigen Trainingsmaterialien werden Sie auch sehen, wie die ST-eigene und mit *ac6-tools* entwickelte *System Workbench for STM32-IDE* (kurz: *SW4STM32*) genutzt wird. Die einfachste Variante ist die auf Eclipse basierte *STM32CubeIDE* [5], die den CubeMX mit dem Atollic-TrueStudio vereint und seit dem Jahreswechsel auch den MP1 unterstützt.

Zwei erwähnenswerte Tools sind noch im CubeMX integriert, die während der Entwicklungsphase mit dem MP1 verwendet werden können:

- Mit dem *Power Consumption Calculator* hat man die Möglichkeit, projektbezogen den Strombedarf in verschiedenen Stromsparmodi hochzurechnen. Dazu gibt es auch einige Beispielanwendungen, die man als Referenz verwenden kann.
- Mit der *DDR Tool Suite* lässt sich die DDR-Speicheranbindung sowohl testen (zum Beispiel als Stress-Test) als auch konfigurieren und optimieren. Stichworte für Kenner der Materie sind dabei *Deskew*, *DQS Gating* oder *Eye Centering*. Nach erfolgreicher Optimierung lassen sich die Konfigurationsdaten auf dem MP1 im *FSBL* (also meist im *TF-A Secure-Bootloader*) speichern. Mit diesem Tool möchte man den Hardware-Entwicklern diese Herausforderung so weit wie möglich abnehmen. Selbstverständlich müssen für die DDR-Tests der MP1 mit der Cube-Umgebung verbunden sein sowie die Boot-Pins in den richtigen Zustand gesetzt werden, damit die nötige Firmware im *SSBL* (siehe übernächster Abschnitt) lädt.

Lücken und Tücken der GPU

Ein Kritikpunkt aus Entwicklersicht ist, dass der Großteil der internen GPU-Bibliotheken nur nach einem erfolgten NDA-Abschluss mit Vivante-Hersteller *VeriSilicon* einzusehen ist. Das kennt man von einigen anderen. Eine transparente Software kann nämlich durchaus eine Bedingung des (End-)Anwenders sein, gerade bei sehr kritischen Anwendungen. Es bleibt zu hoffen, dass sich ST bei Weiterentwicklungen diesem Kritikpunkt annehmen wird.

Für die GPU gibt es zusätzlich zum CL-basierten *vCompiler* einige Tools zur Analyse respektive zur Optimierung:

- Den *vProfiler* kann man auf dem MP1 laufen lassen, um die Performance der GPU in Echtzeit aufzuzeichnen und anschließend auf dem PC mit Hilfe des *vAnalyzer* auszuwerten, auch Funktion für Funktion.
- Mit dem *vShader* können Rendering-Themen wie die Schattierung von dargestellten 3D-Elementen in Vorschau erzeugt und voranalysiert werden. Der *vShader* verfügt über ein GUI und bietet einige 3D-Grundformen zur Auswahl.
- Mit *vTexture* (de-)komprimiert man Bilddateien aus und in die *dxt*- und *etc*-Formate.

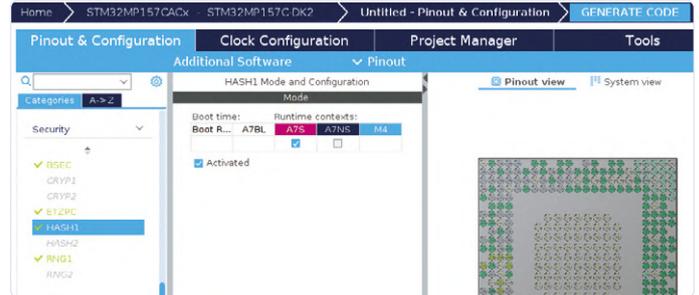


Bild 4. Beispielhaft: die einfache Zuordnung des Hash-Elements zum M4- oder zum M7-Kern (S: secure / NS: non-secure Zone) im CubeMX (Screenshot: Viacheslav Gromov).

Diese Tools finden Sie im *GPU Toolkit*-Package unter [6].

Auf der Produktseite zum MP1 ist auch das *GPU Application Programming Manual PM0263* zu finden, das Tipps gibt, wie man die GPU optimal bei erträglicher Framerate auslastet, auch, um die benachbarten CPUs möglichst von diesem Spiel fernzuhalten. Die Tipps umfassen den richtigen Umgang mit Texturen, *Vertex Buffer Objekten* (VBOs), aber auch Warnungen vor der Nutzung bestimmter GL-Befehle und partieller Löschanweisungen.

Die Software-Troika

Auf dem/den Applikationskern/en werden hauptsächlich zwei Betriebssystemvarianten angeboten, das gängige OpenSTLinux als eine Yocto-Anpassung sowie als Alternative eine Android-Distribution. In **Bild 5** ist das typische Zusammenspiel aller drei Softwarebereiche samt möglicher Bootchain mit einem Linux-Kernel zu sehen. In der Secure-Zone nutzt daneben meist die Sicherheitsumgebung OP-TEE die TrustZone-Eigenschaften. Hierbei startet die sichere Zielumgebung direkt durch den sicheren *First Stage Bootloader* (FSBL), während der Linux-Kernel noch den üblichen U-Boot-Bootloader benötigt. Zu beachten ist insbesondere der gestrichelte Pfad vom *Second Stage Bootloader* (SSBL, also hier U-Boot). Auf diese Weise kann der M4 erfahrungsgemäß circa 600 ms vor dem endgültigen Aktivwerden des Linux-Kernels mit Firmware versorgt und gestartet werden. Selbstverständlich können auf dem M4-Kern anwendungsbedingt sowohl reine Bare-Metal-Lösungen als auch die bekannten Real-Time-Betriebssysteme laufen, wobei das weit verbreitete FreeRTOS von der Cube-Umgebung gut unterstützt wird. Die beschriebene Bootchain ist nur ein Nutzungsbeispiel, denn natürlich kann während der Entwicklung der

AUS DER GERÜCHTEKÜCHE: MEHR STM32MPS

Auf der Embedded World 2020 wurden MP1-(D-/F)-Typen mit einer höheren erlaubten Maximalfrequenz von 800 MHz vorgestellt. Und laut unbestätigten Gerüchten soll in ein paar Jahren auch der MP2 als Weiterentwicklung erscheinen, mit größeren GHz-A-Kernen (und folglich auch größerer Speicher- und Schnittstellenumgebung), effizienterem M-Kern sowie einer noch größeren GPU samt Beschleunigung von Machine-Learning-Berechnungen (vor allem KNNs). Zwischenzeitlich dürfte auch eine reine A7-MPU zu erwarten sein.

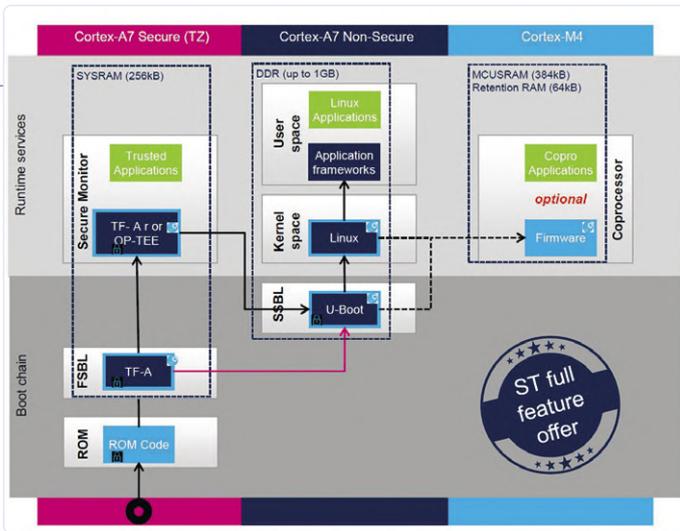


Bild 5. Die Troika: Links die sichere Zone, in der Mitte das applikative Betriebssystem (hier Linux) und rechts der M4 in Echtzeit. Die mögliche Bootfolge mit Calls und Loads ist hier als grobes Flussdiagramm eingezeichnet (Quelle: ST).

M4 auch unabhängig vom A7 durch den JTAG-/SWD-Debugger mit Firmware beladen werden.

Auf die aus M4-Sicht vorhandenen bekannten HAL-, LL- und anderen Bibliotheken muss nicht näher eingegangen werden, aber aus der A7-beziehungswise Linux-Sicht wird es spannend. Der Zugriff auf die möglichen zugeordneten (Peripherie-) Elemente geschieht über die in **Bild 6** genannten Treiber. Daneben gibt es natürlich auch Middleware wie *Wayland-Weston* für GPU-Anwendungen sowie in der möglichen Standardversion die übliche Basis-Middleware wie beispielsweise *apt*, *dpkg*, *python* und *minicom*.

Sie können übrigens in der Online-Wiki [7] mit Development-Bereich nicht nur die OpenSTLinux-Starter, -Developer beziehungsweise Distribution-Pakete sowie die dazugehörige SDK abholen, sondern auch Schritte zur CubeProgrammer-Übertragung auf die MicroSD-Karte und erste Programm-Entwicklungsschritte, Beispiele und „How to's“ für

bestimmte Softwarebestandteile wie das GTK für die Grafik ansehen. Interessant ist das von ST bereitgestellte Package für das in der IoT-Welt bekannte *AWS IoT Greengrass* zur Datenvorverarbeitung „an der Kante“ sowie die Anbindung an die Cloud. Und dann gibt es noch ein Linux-AI-Extension-Paket namens *X-LINUX-AI-CV*, das unter anderem mit dem *TensorFlow Lite* Python-3-Framework auf Applikationsebene Objekterkennung und Bildklassifizierung mit künstlichen neuronalen Netzen (KNNs) erlaubt.

Erste Spielereien

Abhängig vom bootfähigen OS-Image auf der microSD-Karte sollten Sie das Demoprogramm mit einigen Apps sehen, daneben einige Grafik-Demos (wie dem Videoplayer samt mit Bluetooth verbundenen Audiogeräten) und eine Performancesicht auf den über eine IP-Adresse im lokalen WLAN-Netzwerk erreichbaren Monitor namens *netdata*. Um mit der von Ihnen geflashten und bootfähigen microSD-Karte schnell Zugriff auf das Board zu bekommen, können Sie sich mit Hilfe Ihres (virtualisierten) Linux-PC-Systems über die serielle Schnittstelle des integrierten ST-Links mit Tools wie *minicom* verbinden und somit das Terminal des MP1-Linuxsystems sehen und bedienen. Die Alternative wäre eine SSH-Verbindung.

Vom Terminal aus können Sie das Board neu starten und beim Booten durch Drücken einer beliebigen User-Taste in das U-Boot-Terminal gelangen. Von dort aus gibt man den Befehl

```
sums 0 mmc 0
```

so, dass das U-Boot das Dateisystem auf USB-OTG freigibt. Nun können Sie sich vom am OTG angeschlossenen Linux-Rechner aus mit Befehlen wie

```
$mount | grep userfs
```

die einzelnen *File-System*-Bereiche genauer anschauen und beispiels-

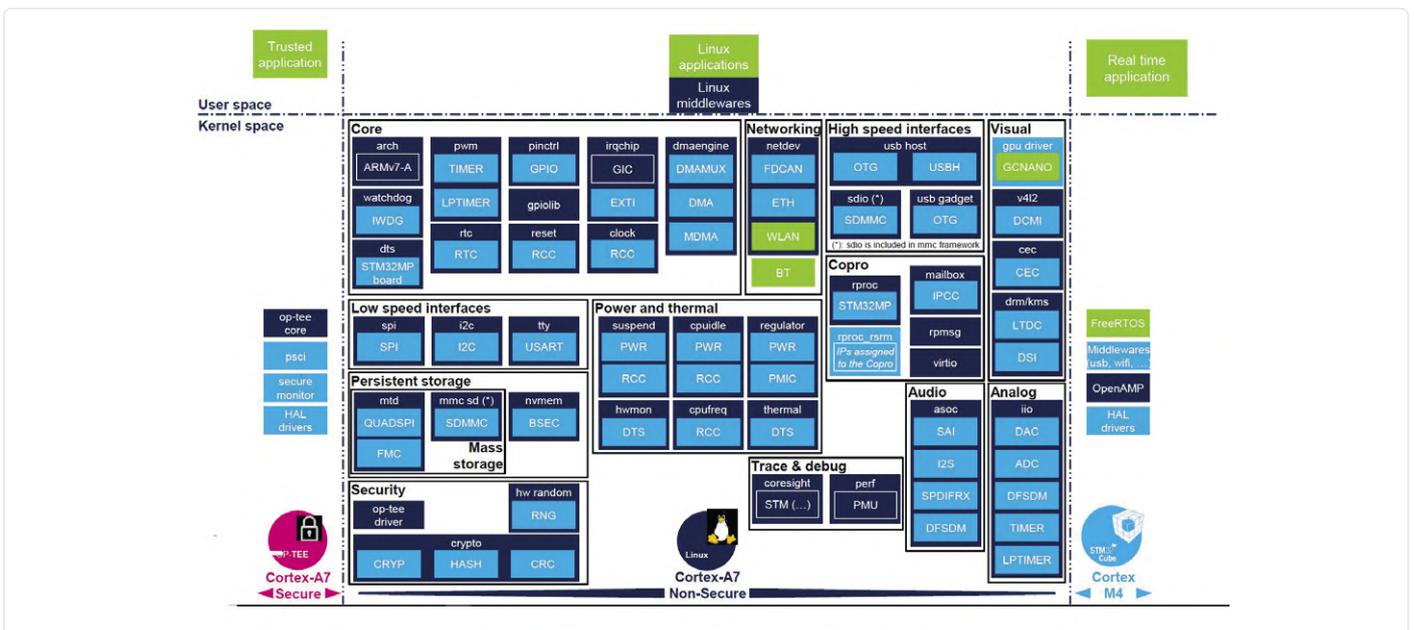


Bild 6. Die „Peripherie“-Treiber in allen drei Zonen unter Linux auf dem applikativen HMP-Teil. Die hellblau markierten Libraries werden von ST bereitgestellt, die dunkelblauen wurden von/mit der Community entwickelt/unterstützt (Quelle: ST).

Listing 1. Ausschnitt aus dem Demoprogramm zur Verdeutlichung des einfachen Umgangs mit der GTK-Grafikbibliothek. Hier ist der Grundaufbau der Seiteninfoleiste sowie eines Icons zu erkennen.

```
def create_page_icon(self):
    page_main = Gtk.HBox(False, 0)
    page_main.set_border_width(0)

    # create a grid of icon
    icon_grid = Gtk.Grid(column_homogeneous=True, row_homogeneous=True)
    icon_grid.set_column_spacing(20)
    icon_grid.set_row_spacing(20)

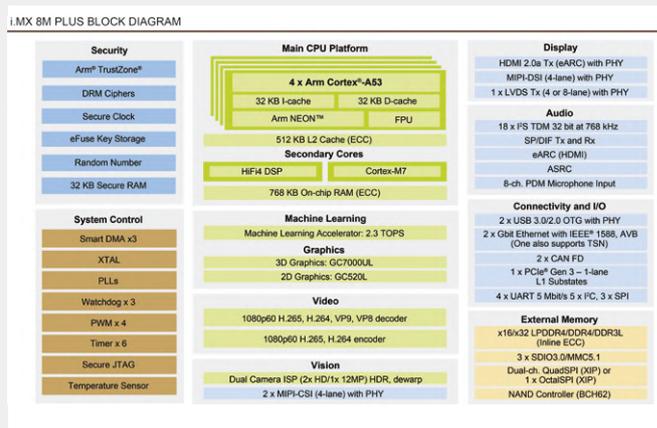
    # STM32MP1 Logo and info area
    logo_info_area = _load_image_Box(self, „%s/pictures/ST11249_Module_STM32MP1_alpha.png“ % DEMO_
        PATH, „%s/pictures/ST13340_Info_white.png“ % DEMO_PATH, self.board_name, -1, 160)
    rgba = Gdk.RGBA(0.31, 0.32, 0.31, 1.0)
    logo_info_area.override_background_color(0, rgba)

    # Button: Netdata icon
    eventBox_webserv = _load_image_eventBox(self, „%s/pictures/netdata-icon-192x192.png“ % DEMO_PATH,
        "netdata", "perf monitor", -1, self.icon_size)
    eventBox_webserv.connect(„button_release_event“, self.wifi_hotspot_event)
    eventBox_webserv.connect(„button_press_event“, self.highlight_eventBox)

    ...
```

BEISPIELE WEITERER EMBEDDED-HMPS

Der Schein sollte nicht trügen, der MP1 ist nicht alleine auf dem aktuellen Prozessormarkt, der auch davor schon einige Embedded-HMPs anbot. Es gibt unzählige anwendungsbedingte Kombinationen und Leistungsgrößen. Die konsumentennahe Embedded-Welt schließt nach oben hin mit den üblichen Smartphone-HMPs ab (abseits von Sonderanwendungen wie Autonomes Fahren). Ein aktuelles Beispiel ist der in der *Galaxy-S10*-Reihe in Europa verkaufte *Exynos 9820 Octacore* mit zwei von *Samsung* selbst entwickelten Kernen (sie heißen M4, haben aber nichts mit ARM-Cortex zu tun!) und zwei A75- und vier A55-Kernen. Im unteren Leistungssegment ist dagegen der *RT1170* von *NXP* angesiedelt. Er besteht aus einem M7- und einem M4-Kern sowie einer 2D-GPU. Streng genommen ist es also kein Prozessor wie die anderen, sondern eher eine doppelte Sonder-MCU mit externem Flashspeicher. Dennoch hat er eine maximale Taktfrequenz von 1 GHz (sic!) und eine würdige Peripherie und Security. *T1* geht unter anderem mit der *Sitara-AM57x*-Familie im leistungsfähigen Markt an den Start. Als Hauptanwendungen wird der Multimedia- und ML-Industriebereich genannt. Dieser HMP hat im Maximalausbau zwei A15-Kerne, zwei DSP-Coprozessoren C66x, zwei Dual-M4-Kerne, eine Dual-3D-GPU, Videobeschleuniger und so weiter. 4K-Auflösung ist mit diesen



Prozessor-Gehilfen also nicht unerreichbar. Der der möglichen Weiterentwicklung STM32MP2 am meisten ähnelnde HMP wurde auf der CES2020 vorgestellt, der *i.MX 8M Plus* von *NXP* (siehe **Blockdiagramm**). Er verfügt abgesehen von den vier A53-Kernen und einem M7-Kern über gleich zwei Grafikeinheiten (2D/3D), einen KNN-Beschleuniger, Zweifach-Kamera-Anschluss und ebenfalls TrustZone- und andere Sonder-Sicherheitsmerkmale.

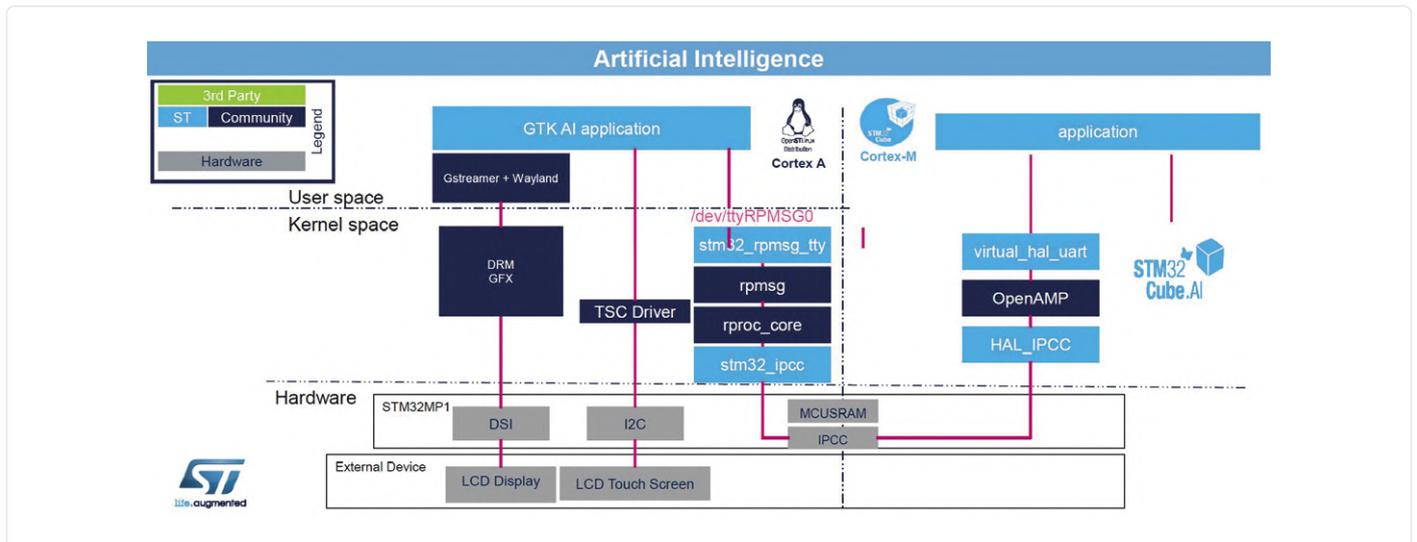


Bild 7. Die Arbeit eines Dreamteams: Die AI-Touch-Handschrift-Anwendung auf dem M4 ist über das OpenAMP beziehungsweise die IPCC-HAL zum RPMsg und bis hin zur GTK-Grafikapplikation auf den A7-Kernen und der GPU verbunden (Quelle: ST).

weise die Demo- oder andere Beispielanwendungen modifizieren. Für die Übertragung und Ausführung von M4-Programmen aus Linux kann man dagegen aus A7-Linuxsicht über die Unterordner von `/sys/class/remoteproc/remoteproc0/` gehen.

Wie ein fertiges Programm aussehen könnte, das beide Kernwelten des HMPs beteiligt, zeigt das Beispiel in **Bild 7**. Dieses KI-Programm ist eine der Apps der Demo und erkennt auf dem Touchdisplay mit dem Finger gezeichnete Buchstaben und Zeichen und führt gegebenenfalls die damit verknüpften Anwendungen aus. Dabei sitzt das trainierte und durch CubeAI portierte neuronale Netz im M4-Bereich, während die ganze Grafik-HMI natürlich auf dem A7 beziehungsweise der GPU läuft. Das Zusammenspiel auch mit der erwähnten IPCC-Schnittstelle und dem Shared SRAM durch das bekannte *OpenAMP*-Framework ist dabei gut zu beobachten.

Das User-Interface auf dem Display kann durch die *GTK*-Bibliothek sehr schnell verwirklicht werden, wie das **Listing 1** anschaulich beweist. Natürlich gibt es auch andere Grafik-Frameworks für unterschiedlichste Funktionen wie *GStreamer* oder *Wayland/Weston*, und selbst *QT* kann als Metaebene eingebunden werden.

Ob Einsteiger, Umsteiger, Cortex-M- oder Linux-Kenner: Mit dieser

kleinen Markt- und Dokumentationsübersicht haben wir Ihnen einen kleinen Stoß in die für Sie noch fremde HMP-Welt gegeben. Jetzt ist der Punkt gekommen, an dem wir Sie guten Gewissens entlassen können, auf dass Sie sich weiter in der HMP-Welt umtun! 

200072-01



IM ELEKTOR-STORE

- **Pyboard D-series with STM32F722 and WiFi/BT:**
www.elektor.de/pyboard-d-series-with-stm32f722-and-wifi-b
- **Buch „Programming with STM32 Nucleo Boards“:**
www.elektor.de/programming-with-stm32-nucleo-boards
- **STM32 Nucleo L476RG Board:**
www.elektor.de/stm32-nucleo-l476rg-board

WEBLINKS

- [1] **Produktseite der MPU-Familie:** www.st.com/en/microcontrollers-microprocessors/stm32mp1-series.html
- [2] **STM32MP1-Onlinetraining:**
www.st.com/content/st_com/en/support/learning/stm32-education/stm32-online-training/stm32mp1-online-training.html
- [3] **Discovery Kit DK2 mit STM32MP157C:** <https://bit.ly/37HXOF8>
- [4] **STM32CubeMP1-Paket:** <https://bit.ly/2UaBhwX>
- [5] **STM32CubeIDE:** www.st.com/en/development-tools/stm32cubeide.html
- [6] **STM32MP157 GPU Toolkit:** www.st.com/en/development-tools/stm32mp157gputk.html
- [7] **Wiki/Development Zone mit OS-Downloadanleitungen:** <https://wiki.st.com/stm32mpu>

Testing Rig für 16F18877 und ähnliche PICs

Von **Tam Hanna**

Der PIC16F18877 ist definitiv nicht das preiswerteste Bauteil. Andererseits hat er ausreichend Speicher, eine komplexe interne Architektur und viele Peripheriegeräte, die uns Experimente ermöglichen.

Bei der Arbeit mit Embedded-Systemen ist es empfehlenswert, einen soliden Testharnisch zusammenzubauen. Wer einen Softwarefehler sucht und am Ende feststellt, dass es sich um ein loses Kabel auf der Steckplatine handelt, flucht mehrsprachig. Zudem sind Steckplatinen ob der hohen Kapazitäten und der kaum kontrollierbaren Leitungseinduktivitäten sowohl für schnelle als auch für präzise Schaltungen nur wenig geeignet. **Bild 1** zeigt die Grundschialtung, die wir für Experimente benötigen.

Sie können das Testing Rig grundsätzlich aufbauen, wie Sie wollen. Ich habe diverse im Labor herumliegende Platinensubstrate genutzt, um das Ergebnis zu realisieren (**Bild 2**). Dass das nicht unbedingt sehr komfortabel ist, sei hier – schon im Interesse der Fairness – angemerkt, eine doppelseitige Lochrasterplatine hätte viel Löterei eingespart. Wer mit Platinenlayoutsoftware gut umgehen kann, sollte sich einen (chinesischen) Anbieter von Planaren suchen - der Autor zahlt für die meisten Zweiläger weniger als 50 €.

PICs sind in Sachen Versorgungsspannung anspruchslos: Wir setzen aus Gewohnheitsgründen auf 5 V. Wenn Sie stattdessen mit 3,3 V arbeiten möchten, ist dies allerdings auch kein Problem. Der Schaltungsaufwand entsteht durch die ICSP-Programmierschaltung. Es handelt sich dabei um eine Gruppe von Microchip-spezifischen Pins, die eine Kommunikation mit dem Computer zwecks Programmierung und Debugging erlauben.

Praktische Designs exponieren die ICSP-Pins auch bei den Platinen von ausgelieferten Geräten: Bei kleinen Fertigungsmengen ist es billiger, die Platinen von Hand zu programmieren, als Microchip zur Anlieferung eines vorprogrammierten PICs zu animieren. Zudem reduziert die Verfügbarkeit von ICSP die Kosten für Firmware-Upgrades nach der Auslieferung. Logisch ist, dass sich Leute, die sich für den Code interessieren, ebenfalls über ICSP-Pins freuen. In der Praxis spielt dies allerdings keine große Rolle: Es gibt in der ehemaligen Sowjetunion Hunderte von Labors, die PICs für weniger als 1000 Euro pro Stück auslöten, dekupieren und auslesen.

Wichtig ist der 50-k Ω -Widerstand zwischen dem MCLR-Pin und der Versorgung. MCLR ist ein Aktiv-Low-Pin, der den PIC zum Resetten zwingt. Wer einen Widerstand mit ausreichend hohem Wert anschließt, kann den Pin „nach oben ziehen“. Das Programmiergerät leitet den minimalen Strom nach Masse ab, wenn während der Kommunikation ein Reset auszulösen ist.

Wer davon ausgeht, dass der Harnisch ein langes Leben hat, sollte einen Textool-Sockel verwenden. Unter [aliexpress.com](https://www.aliexpress.com) bekommt man die Teile durchaus günstig, muss aber mit Lieferzeiten von bis zu vier Wochen rechnen. Meine Wenigkeit umgeht dieses Problem, indem er immer drei oder vier auf einmal kauft. 

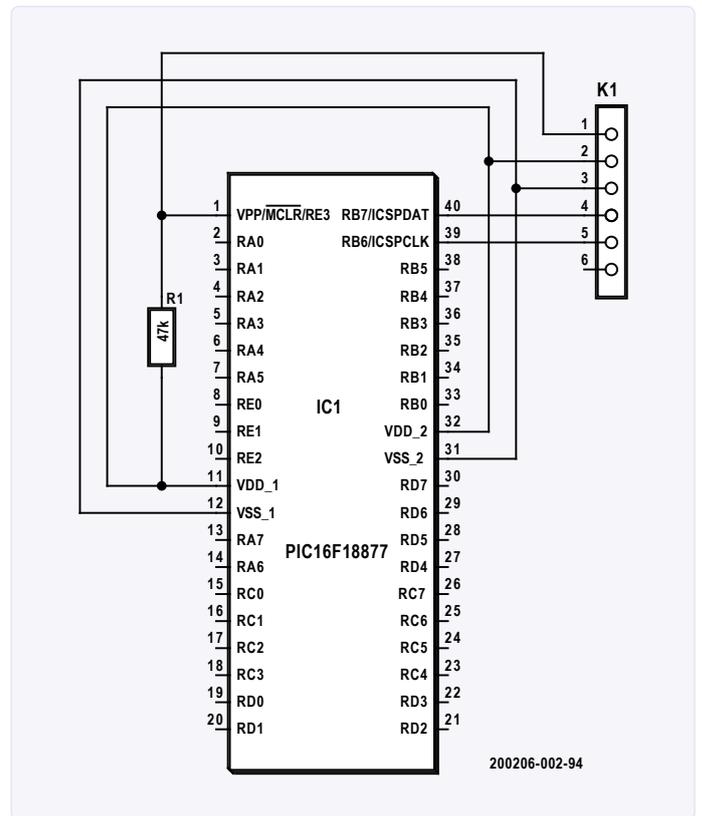


Bild 1. Der sechspolige Header ermöglicht die Kommunikation mit dem Programmiergerät.

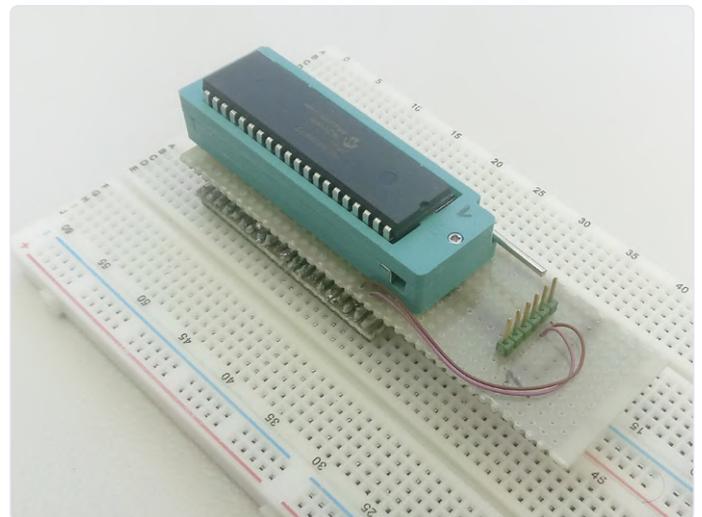


Bild 2. Dank der Pins auf der Unterseite kann das Testing Rig auf einem Steckbrett befestigt werden.

200206-02

Listing 2. SPI-Slave.

```
'-----  
'UNO_spi3.BAS SPI Slave  
'-----  
$regfile = "m328pdef.dat"  
$crystal = 16000000  
$baud = 9600  
  
Dim Addr As Byte  
Dim B As Bit  
Dim Dout As Word  
Dim Din As Word  
Dim N As Byte  
Dim I As Byte  
  
S1 Alias Pinc.0  
Portc.0 = 1  
S2 Alias Pinc.1  
Portc.1 = 1  
Sck Alias Pinb.5  
Portb.5 = 1  
Mosi Alias Pinb.3  
Portb.3 = 1  
  
Cs Alias Pinb.2  
Portb.2 = 1  
...  
Do  
  Do  
    Loop Until Cs = 0  
    Din = 0  
    For N = 1 To 10  
      Shift Din , Left  
      Do  
        Loop Until Sck = 1  
        Din = Din + Mosi  
      Do  
        Loop Until Sck = 0  
    Next N  
  Do  
    Loop Until Cs = 1  
    Locate 1 , 1  
    Lcd Din  
    Lcd " "  
    Print Din  
  Loop  
End
```

Einsatz, die Chip-Select-Leitung /CS. Der Strich / steht dafür, dass diese Leitung im aktiven Zustand low ist. Die CS-Leitung kann eingesetzt werden, um mehrere Slaves an einen Master zu hängen. Sie teilen sich dann die Datenleitung und die Taktleitung, bekommen aber jeweils eine eigene CS-Leitung. Wenn diese low ist, dann weiß der entsprechende Slave, dass er gemeint ist. Die Sache hat aber auch noch einen zweiten Nutzen. Wenn der Slave erst mit Verspätung eingeschaltet wurde, könnte es zu einem Missverständnis kommen, welche Bits gerade übertragen werden. Wenn er aber erstmal abwartet, bis seine CS-Leitung von high auf low gezogen wird (fallende Flanke), dann weiß er, dass die Übertragung beginnt. Und sollte einmal ein Störpuls als Taktsignal gelesen werden, sind zwar die folgenden Daten nur noch Müll, beim nächsten Zugriff ist aber wieder alles korrekt.

Der ATmega328 verwendet den SPI-Bus auch zu seiner eigenen Programmierung über ein externes Programmiergerät. Am sechspoligen Programmierstecker auf dem Arduino und dem Elektor Extension-Shield [1] (ICSP im Schaltplan) findet man daher

- die Taktleitung SCK (Serial Clock, B5),
- die Datenleitung zum Schreiben MOSI (Master out Slave in, B3)
- und die Datenleitung zum Lesen MISO (Master in Slave out, B4).

Eine CS-Leitung gibt es nicht, aber Reset tritt an ihre Stelle, denn der Programmiervorgang findet bei heruntergezogener Reset-Leitung statt. Diese Leitungen sollen bei uns nun in genau der gleichen Funktion verwendet werden. Das hat den Vorteil, dass man bei Gelegenheit auch die Hardware-SPI-Einheit des Controllers verwenden kann. Dann braucht man nicht mehr jedes Bit einzeln auf die Leitung legen (wie bisher mit den in den Listings gezeigten Codezeilen); und alles geht noch schneller. Doch eine CS-Leitung muss noch zusätzlich her, hier wurde dafür die Leitung B2 ausgewählt.

Der Master (**Listing 1**) verwendet MOSI als Ausgang und erzeugt den Takt und das CS-Signal. Der Ablauf wurde durch drei Verzögerungen von je einer Millisekunde etwas verlangsamt, sodass man die Signale bequem mit dem Oszilloskop ansehen kann. Außerdem soll die Sache für den Slave nicht zu schwierig werden. Wer möchte, kann ja mal die Grenzen ausloten und die Verzögerungen so weit herabsetzen, bis es zu Übertragungsfehlern kommt.

Für den Slave (**Listing 2**) sind die drei Leitungen Eingänge. Er wartet jeweils auf bestimmte Flanken an CS und SCK und liest dann ein Bit an MOSI ein. Weil hier alles per Software erledigt wird, muss jede Flanke in einer Do-Loop-Schleife abgewartet werden. Dafür wird etwas Zeit benötigt, sodass die Übertragung langsamer sein muss als mit einem Hardware-SPI. Dreht man am Poti, dann wird die Änderung beim Slave auf dem LCD des Extension-Shields sichtbar. ◀

200202-04

WEBLINK

- [1] **Mein erstes Shield :-):**
www.elektormagazine.de/magazine/elektor-201407/26859

Sechs Oszillatoren

Von **Burkhard Kainka**

Wenn man immer nur bereits vorhandene Signale verarbeiten könnte, wäre die Elektronik um einen Aspekt ärmer: Das Erzeugen von Schwingungen - scheinbar aus dem Nichts. Oszillatoren sind wichtige Bestandteile vieler Geräte und besitzen die unterschiedlichsten Anwendungen. So lassen sich Töne erzeugen, aber auch Testsignale für Schaltungen und Baugruppen.

RC-Oszillatoren

Jeder kennt das unangenehme Pfeifen, das in Mikrofonanlagen entstehen kann. Das Phänomen tritt bei einer akustischen Rückkopplung zwischen Lautsprecher und Mikrofon auf. Es können Töne unterschiedlicher Höhe entstehen, die sich nur durch einen größeren Abstand der Anlagenteile oder eine geringere Verstärkung verhindern lassen. Prinzipiell führt jede ausreichend starke Rückkopplung zu Schwingungen. Die Rückkopplung kann auch rein elektrisch erfolgen, indem ein Ausgangssignal an den Eingang zurückgekoppelt wird. Bedingung ist die richtige Phasenlage, wie sie bei einem zweistufigen Verstärker gegeben ist. Die Schaltung in **Bild 1** ähnelt der eines Multivibrators, allerdings mit einstellbarer Rückkopplung. Während der Multivibrator immer Rechtecksignale liefert, können hier auch Sinussignale oder andere Kurvenformen entstehen. Über den Lautstärkereglers lässt sich die Rückkopplung so einstellen, dass gerade schwache Schwingungen entstehen. Diese sind dann meist sinusförmig.

Aber es geht auch mit einem einzelnen Transistor, obwohl dieser die Phase um 180 Grad dreht. Die erforderliche zusätzliche Phasendrehung um 180 Grad lässt sich auch mit mehreren hintereinander geschalteten RC-Gliedern erreichen. Der Phasenschieber-Oszillator in **Bild 2** erzeugt ein Sinussignal von ca. 800 Hz, ideal ist dieser als Morse-Übungs-generator oder als Signalgenerator für den Test von NF-Verstärkern. Auch mit einem Feldeffekttransistor BS170 funktioniert der Phasenschieber-Oszillator prima. Die Schaltung in **Bild 3** wurde sehr hochohmig ausgelegt und liefert eine Frequenz von 10 Hz. Es wird hier nur ein Betriebsstrom von ca. 30 μ A benötigt.

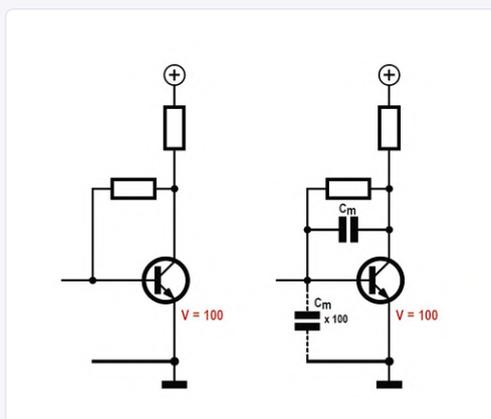
Ringoszillatoren

Bisher wurden eine Stufe oder zwei Stufen verwendet. Aber was ist mit drei Stufen in Emitterschaltung? Man würde eigentlich erwarten, dass aus der Rückkopplung eine Gegenkopplung wird, denn insgesamt wird die Phase um 180 Grad gedreht. Tatsächlich aber entstehen Schwingungen (**Bild 4**). Die erzeugte Frequenz steigt mit der Betriebsspannung und kann bis zu 1 MHz erreichen. Was passiert hier? Man hat einen gegengekoppelten dreistufigen Verstärker mit sehr hoher Spannungsverstärkung. Jede der drei Stufen verursacht aber gleichzeitig eine kleine Zeitverzögerung. Alle drei Verzögerungen zusammen führen auf einer ganz bestimmten Frequenz zu einer zusätzlichen Phasenverschiebung von noch einmal 180 Grad. Für diese Frequenz wird dann aus der Gegenkopplung eine

Der Millereffekt

Ein Verstärker in Emitterschaltung hat eine typische Spannungsverstärkung von etwa 100fach. Das gilt bis zu recht hohen Frequenzen (allerdings manchmal nicht so hoch wie man es sich wünschen würde). Während die Transitfrequenz eines BC547 bei etwa 300 MHz liegt (bei 300 MHz ist der Stromverstärkungsfaktor auf 1 gesunken), ist die Grenzfrequenz dieser Verstärkerschaltung sehr viel geringer. Das gilt besonders, wenn die Widerstände in der Schaltung recht hochohmig ausgelegt sind. Schuld daran sind die internen Sperrschichtkapazitäten des Transistors.

Besonders wirksam ist die Basis-Kollektor-Kapazität C_{BC} , obwohl sie beim BC547 nur etwa 5 pF beträgt. Das liegt am Miller-Effekt. Der Miller-Kondensator $C_m (= C_{BC})$ zwischen Eingang und Ausgang des invertierenden Verstärkers wird nämlich von zwei Seiten aus umgeladen. Steigt z.B. die Basisspannung um 1 mV, dann sinkt die Kollektorspannung gleichzeitig um 100 mV. Es muss also etwa die 100-fache



Ladung aufgebracht werden. Am Eingang liegt scheinbar ein Kondensator, der um den Spannungsverstärkungsfaktor größer ist als die Miller-Kapazität. In diesem Fall wären das etwa 500 pF. Je nach Innenwiderstand der angeschlossenen Signalquelle bildet sich damit ein Tiefpassfilter, das einem die hohe Grenzfrequenz verhagelt.

Für einen Verstärker heißt das: Wenn es auf eine hohe Grenzfrequenz ankommt, sollte man die Schaltung möglichst niederohmig

dimensionieren. Außerdem kann es im Einzelfall sinnvoll sein, mit einer geringeren Spannungsverstärkung zu arbeiten, z.B. indem man den Ausgangswiderstand klein macht. Außerdem kann man mit speziellen HF-Transistoren viel erreichen, die sehr viel geringere Sperrschichtkapazitäten haben. Für einen Oszillator bedeutet die Miller-Kapazität, dass man einen Oszillator ohne frequenzbestimmende Kondensatoren bauen kann, denn diese bringt der Transistor schon selbst mit.

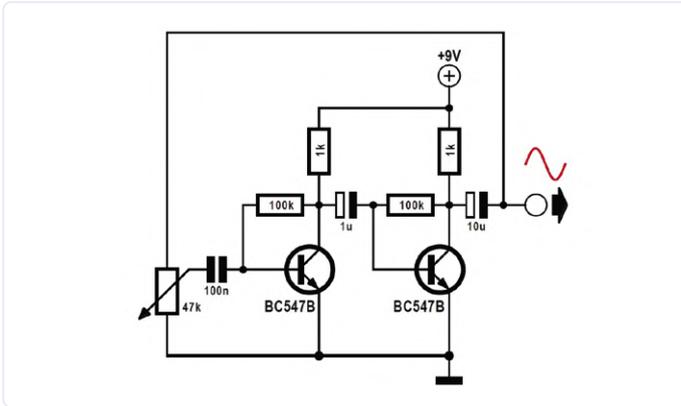


Bild 1. Schwingungserzeugung durch Rückkopplung.

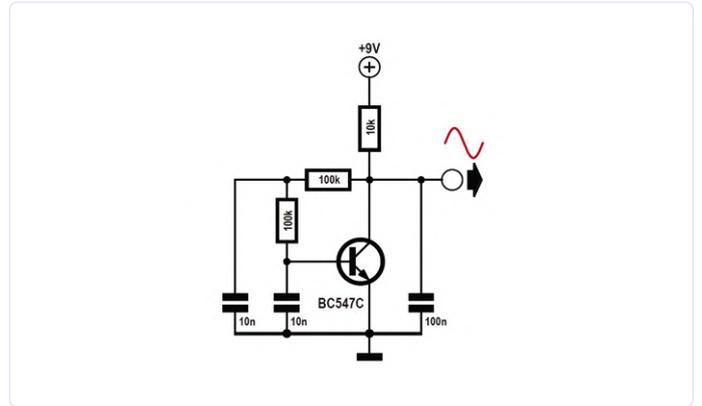


Bild 2. Oszillator mit Phasenschieber.

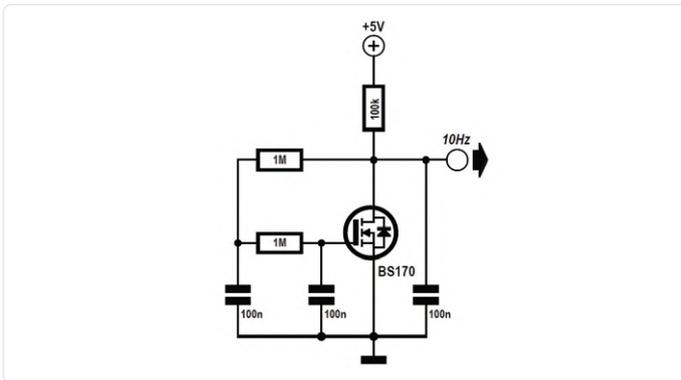


Bild 3. Phasenschieberschwingungserzeugung mit FET.

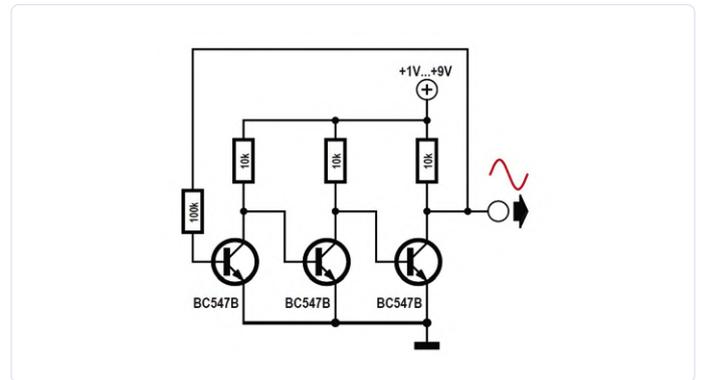


Bild 4. Ein Oszillator ohne Kondensatoren.

Rückkopplung. So entstehen Schwingungen. Will man eine solche Schaltung dagegen nicht als Oszillator, sondern als empfindlichen Verstärker einsetzen, muss man sich alle Mühe geben, eine mögliche Rückkopplung zu verhindern. Bei so hoher Verstärkung ist es nicht ganz einfach, Eigenschwingungen zu unterbinden.

Einen dreistufigen Oszillator zu bauen ist einfacher als einen dreistufigen Verstärker. Je kleiner der mittlere Kollektorstrom ist, desto höher wird die Schaltung, und desto stärker wirken sich die internen Kapazitäten des Transistors aus. Deshalb wird bei kleiner Betriebsspannung die Zeitverzögerung größer und die erzeugte Frequenz kleiner.

Eine solche Schaltung bildet einen Ring aus einzelnen Verstärkerstufen und wird deshalb auch als Ringoszillator bezeichnet. Er funktioniert auch mit fünf, sieben oder neun Stufen. Die Bedingung ist nur, dass für Gleichspannung eine Gegenkopplung besteht. Bei einer geraden Anzahl von Stufen würde dagegen immer ein statisches Flipflop entstehen. Ein dreistufiger Ringoszillator kann sehr hochfrequent und damit sehr sparsam betrieben werden. Mit drei Kollektorwiderständen von 1 M Ω läuft der Oszillator schon mit 0,5 V und braucht weniger als 1 μ A. Deshalb reicht sogar schon eine Fotodiode BPW34 im Sonnenlicht als kleine Solarzelle, um den Oszillator zu versorgen (**Bild 5**). Am Ausgang

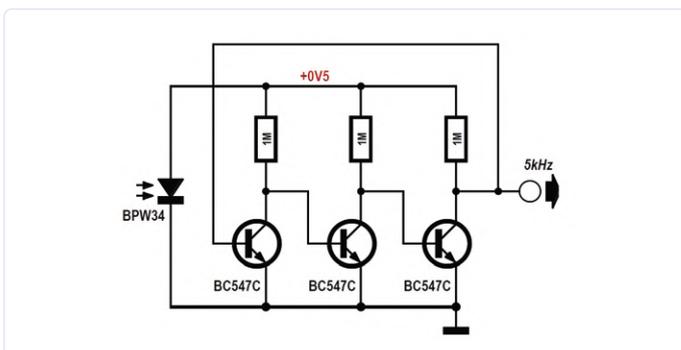


Bild 5. Ringoszillator mit Solarzelle.

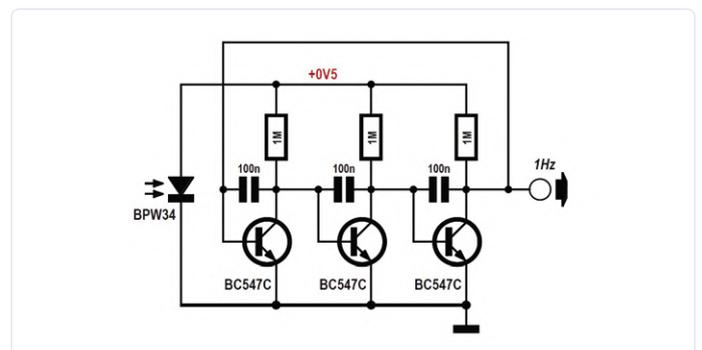


Bild 6. Langsamer und sparsamer Ringoszillator.

findet man eine Frequenz von ca. 5 kHz. Die Frequenz steigt mit der Beleuchtung, vielleicht könnte man die Schaltung sogar sinnvoll als Lichtsensor einsetzen.

Nur 5 kHz, und das ganz ohne Kondensatoren? Seltsam, wo doch ein Transistor nur wenige Pikofarad hat. Hier wirkt sich die sogenannte Millerkapazität (siehe Kasten) aus, welche die Kollektor-Basiskapazität um den

Faktor der Spannungsverstärkung vergrößert an den Eingang legt. Nach diesem Prinzip kann man leicht noch eins drauflegen und mit zusätzlichen Kondensatoren zwischen Basis und Kollektor eine sehr tiefe Frequenz erzeugen (**Bild 6**). Mit dreimal 100 nF kommt man nun auf ca. 1 Hz. ◀

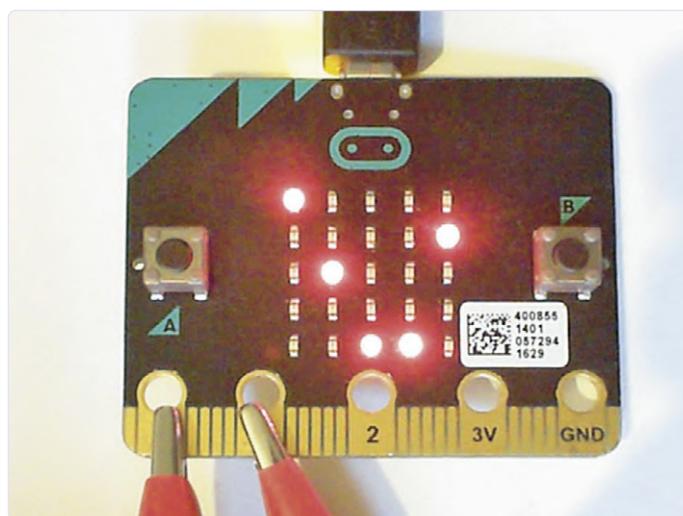
200205-04

Absolutes Minimal-Oszilloskop mit LED-Display

Von **Burkhard Kainka**

Ein ganz einfaches Oszilloskop ist besser als keins, und manchmal kommt es mehr darauf an, dass das Gerät sehr klein, autonom und handlich ist. Hier werden die Messdaten grafisch auf dem LED-Display eines BBC micro:bit mit 5 mal 5 LEDs dargestellt (**Listing 1**). Auch wenn man durch ein sehr viel besseres Gerät verwöhnt ist, kann man durchaus damit arbeiten. Es ist ganz erstaunlich, was mit einem so einfachen Oszilloskop noch erkennbar ist.

Das Mini-Oszilloskop verwendet den Port 1 als analogen Eingang und nutzt zusätzlich den Port 0 als PWM-Ausgang. Bei einer Periode von 500 µs entsteht ein Ausgangssignal mit einer Frequenz von 2 kHz.



Listing 1. 5x5-LED-Oszilloskop mit BBC micro:bit.

```
//LED-Scope
#include "MicroBit.h"
MicroBit uBit;
int main()
{
    int y;
    uBit.init();
    uBit.io.P0.setAnalogValue(512);
    uBit.io.P0.setAnalogPeriodUs(500);
    uBit.display.enable();
    MicroBitImage image(5,5);
    while (1) {
        for(int x = 0; x < 5; x++){
            y = 4- (uBit.io.P1.getAnalogValue()/205);
            image.setPixelValue(x,y,255);
        }
        uBit.display.print(image);
        uBit.sleep(500);
        image.clear();
    }
}
```

Eine direkte Verbindung zum Messeingang zeigt die Grenzen des AD-Wandlers. Die Abtastzeit ist offensichtlich zu lang, um scharfe Flanken des PWM-Signals abzubilden. Die Grenzfrequenz des einfachen Oszilloskops liegt also irgendwo unter 10 kHz. Das reicht zwar noch nicht so ganz für das HF-Labor, wohl aber für viele einfache Messungen und Experimente. ◀

200204-04



IM ELEKTOR-STORE

- ▶ **E-Book „BBC micro:bit“**
www.elektor.de/bbc-micro-bit-e-book
- ▶ **JOY-iT BBC micro:bit Go Set**
www.elektor.de/joy-it-bbc-micro-bit-go-set
- ▶ **Bread:bit edge connector breakout board for BBC micro:bit**
www.elektor.de/bread-bit-edge-connector-breakout-board-for-micro

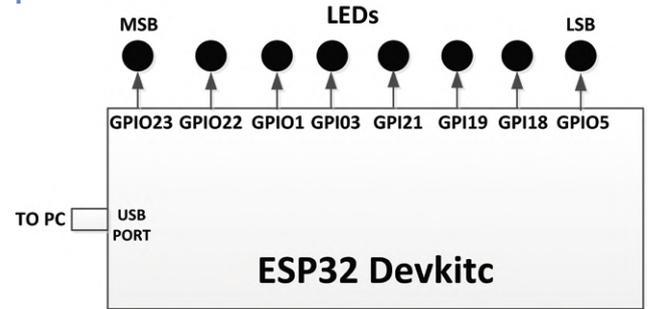
Knight Rider: LED-Lauflicht mit dem ESP32

Von **Dogan Ibrahim**

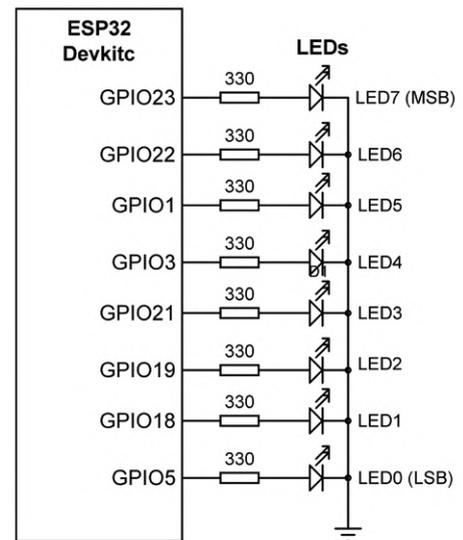
Knight Rider war in den 1980ern ein TV-Actionfilm mit einem superintelligenten Auto namens KITT, das spricht und selbst navigiert. Die Serie wurde durch den Lichteffekt an der Vorderseite des Autos und dem damit verbundenen *Zoomzoom*-Sound unsterblich. Die Lichter gehen nacheinander in beide Richtungen an und wieder aus, und diese Bewegung wiederholt sich kontinuierlich.

In diesem kleinen Projekt sind acht LEDs an ein ESP32-DevKitC angeschlossen und simulieren die Knight-Rider-Autolichter. Das Blockschaltbild des Projekts in **Bild 1** und der Schaltplan in **Bild 2** zeigen die acht LEDs, die über 330- Ω -Strombegrenzungswiderstände mit den GPIO-Ports verbunden sind. Das Projekt lässt sich leicht auf einer Lochrasterplatine aufbauen (**Bild 3**). Schließlich zeigt **Listing 1** das PDL-Script (Project Description Language / Pseudocode) des Projekts. Das eigentliche **Listing 2** ist die praktische Umsetzung. Zu Beginn des Programms wird ein Array namens *LEDs* eingerichtet, um die Portnummern der LEDs zu speichern. Dann werden die GPIO-Ports, an die die LEDs angeschlossen sind, als Ausgangsports konfiguriert.

1



2



Listing 1. Knight Rider Pseudocode.

BEGIN

Store LED port numbers in array LEDs

Configure LED port pins as outputs

DO FOREVER

Do k From 0 to 8

Turn ON LED at index LEDs[k]

Wait 100 ms

Turn OFF LED at index LEDs[k]

ENDDO

DO k From 6 to 0

TURN ON LED at index LEDs[k]

Wait 100 ms

Turn OFF LED at index LEDs[k]

ENDDO

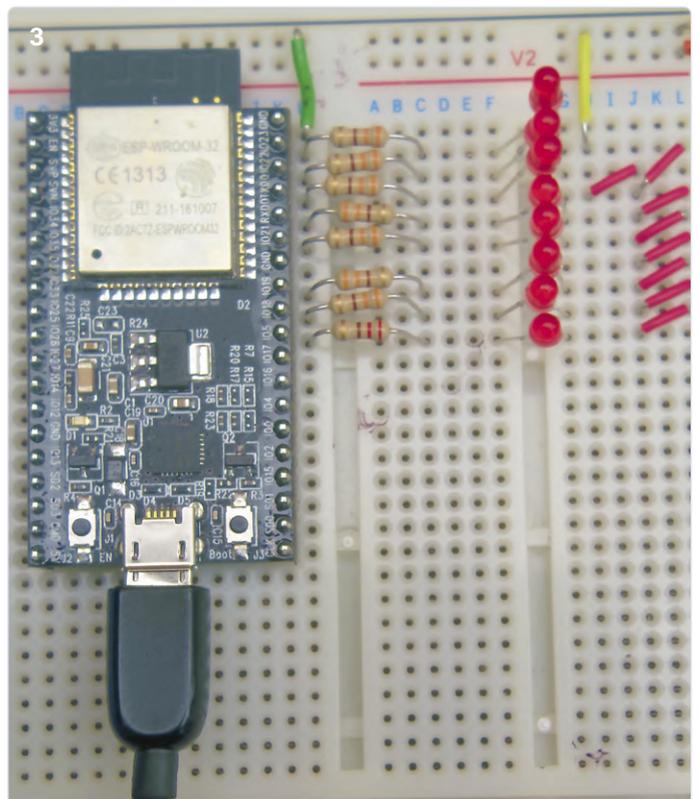
ENDDO

END

WEBLINK

- [1] **Programmdownload:**
www.elektormagazine.de/200234-03

3



Listing 2. ESP32-Code für den Knight Rider.

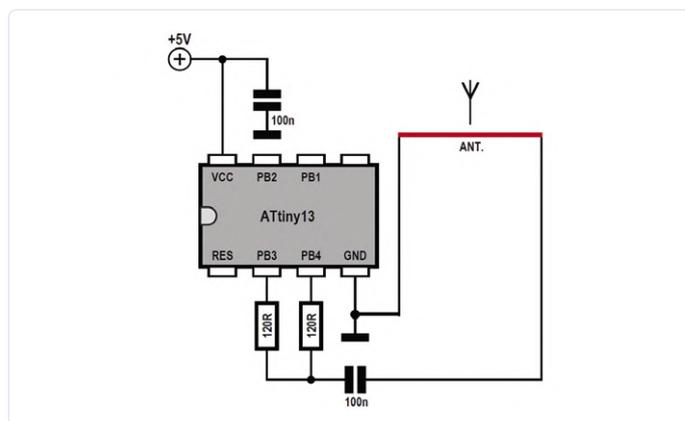
```
/******  
 *           KNIGHT RIDER LEDs  
 *           =====  
 *  
 * In this program 8 LEDs are connected to port  
 * pins GPIO23, GPIO22, GPIO1, GPIO3, GPIO21,  
 * GPIO19, GPIO18, and GPIO5 of the ESP32 DevKitC.  
 * The program simulates the lights of  
 * the Knight Rider car as in the TV action movie  
 * Knight Rider.  
 *  
 * Program: KnightRider  
 *  
*****/  
int LEDs[] = {23, 22, 1, 3, 21, 19, 18, 5};  
unsigned char Count = 0;  
unsigned char del = 100;  
  
//  
// Set GPIO pins 23,22,1,3,21,19,18,5 as outputs  
//  
void setup()  
{  
  unsigned char i;  
  for(i=0; i < 8; i++)  
  
  {  
    pinMode(LEDs[i], OUTPUT);  
  }  
  
  //  
  // Turn the LEDs ON/OFF to simulate the  
  // Knight Rider car  
  //  
  void loop()  
  {  
    for(int k = 0; k < 8; k++)  
    {  
      digitalWrite(LEDs[k], HIGH);  
      delay(del);  
      digitalWrite(LEDs[k], LOW);  
    }  
  
    for(int k = 6; k > 0; k--)  
    {  
      digitalWrite(LEDs[k], HIGH);  
      delay(del);  
      digitalWrite(LEDs[k], LOW);  
    }  
  }  
}
```

Innerhalb des Hauptprogramms werden zwei **FOR**-Schleifen eingerichtet. In der ersten werden die LEDs von MSB bis LSB für 100 ms eingeschaltet und in der zweiten LSB zu MSB für 100 ms eingeschaltet. Der resultierende Effekt ist, dass sich LEDs gegenseitig in beide Richtungen „jagen“.

Das Programm ist als kostenloser Download erhältlich [1]. Als Auftrag, um Ihre Fertigkeiten zu testen: Die Verzögerungszeit zwischen den einzelnen Ausgängen ist auf 100 ms eingestellt. Versuchen Sie, diese Zeit zu modifizieren und sehen Sie sich die Effekte auf dem Display an. **◀**

200234-03

LW/MW-AM-Signalgenerator mit ATtiny13



Von **Burkhard Kainka**

Jeder, der Funkempfänger entwirft, baut oder repariert, kann einen kleinen AM-Signalgenerator gut gebrauchen. Mit einer geeigneten Wellenform – der Generator muss mit einer ausreichend niedrigen Frequenz arbeiten und genügend Oberwellen erzeugen – braucht man sogar nicht einmal die Frequenz einstellen zu können. Der hier beschriebene AM-Signalgenerator mit einem ATtiny13 erzeugt kurze Impulse mit einer Wiederholfrequenz von 70 kHz. Dies führt zu starken Oberwellen, die das gesamte Lang- und Mittelwellen-Spektrum abdecken. Der Impulszug wird in regelmäßigen Abständen kurz unterbrochen, um eine Amplitudenmodulation (AM) mit einer Frequenz von etwa 750 Hz zu simulieren. Ein Radio kann daher das AM-Testsignal unter 70 kHz, 140 kHz, 210 kHz und so weiter empfangen und demodulieren, um einen hörbaren Ton zu erzeugen.

Listing 1. ATtiny13 AM-Generator.

```
'ATtiny13 AM Generator
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4
Config Portb = Output
Dim N As Byte
Do
```

```
For N = 1 To 50      'generate 70 kHz
  Portb = 255
  Portb = 0
Next N
For N = 1 To 50    'pseudo AM 750 Hz
  nop
  nop
Next N
Loop
End
```

Eine Drahtschleife mit einem Durchmesser von etwa 10 cm ergibt eine geeignete Antenne. Sie erzeugt ein magnetisches Wechselfeld, das direkt in den Ferritstab des Empfängers einkoppelt. Der Signalgenerator kann auch für vergleichende Empfindlichkeitsmessungen verwendet werden, indem getestet wird, wie weit vom Generator entfernt das

Signal noch hörbar ist. Ein guter Empfänger sollte ein klares Signal in einer Entfernung vom Sender von bis zu 1,5 m empfangen. Der ATtiny13-Quellcode ist in **Listing 1** zu sehen. ◀

200197-04

Minimalistisches Dipmeter

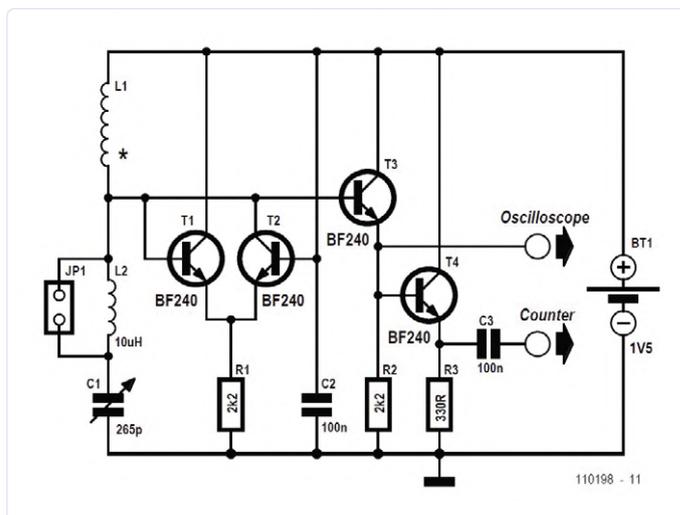
Von **Burkhard Kainka**

Ein Amateurfunker hatte früher in seinem Heimlabor immer ein Dipmeter in Reichweite. Heutzutage kann man sich auch ein Oszilloskop leisten, so dass das Dipmeter an Bedeutung verloren hat und oft nicht mehr vorhanden ist. Eigentlich ist das schade, denn manches geht mit einem Dipmeter viel einfacher. Wer sich dafür (wieder) interessiert, der kann es mit dieser einfachen und ohne weiteres ausreichenden Schaltung ganz schnell selbst bauen. Die interessante Frage ist nämlich: Was braucht man wirklich?

- > Ein Zeigerinstrument zur Anzeige des Dips? Braucht man nicht, das Oszilloskop kann diese Aufgabe übernehmen.
- > Eine große Frequenz-Skala? Braucht man nicht, dafür kann ein Frequenzzähler angeschlossen werden.
- > Wechselbare Spulen? Braucht man auch nicht, die Bereichsumschaltung kann mit einem Jumper erfolgen.

Die Sensorspule L1 hat zehn Windungen und wird auf eine Mignon-Batterie gewickelt. Damit überstreicht man den Bereich von 6 MHz bis 30 MHz. Bei nicht gestecktem Jumper JP 1 liegt noch einmal eine Festinduktivität mit 10 µH in Reihe. Der Frequenz-Messbereich beträgt dann 2,5 MHz bis 10 MHz.

Bei der Messung hält man einen Schwingkreis nahe an die Sensorspule. Beim langsamen Abstimmen mit dem Drehko C1 erkennt man die Resonanzfrequenz daran, dass die Oszillatoramplitude etwas abnimmt. Die Frequenz kann dann direkt am Oszilloskop abgele-



sen werden. Wenn die Frequenz sehr genau angezeigt werden soll, kann man zusätzlich an den zweiten Ausgang einen Frequenzzähler anschließen. ◀

200199-03

Preiswerter E-Scooter

Was taugt ein „zugelassener“ 300-€-E-Tretroller von Lidl?

Von **Dr. Thomas Scherer**

In vielen Städten sind neuerdings überall diese kleinen elektrifizierten Tretroller zu sehen. Man kann sie mieten und natürlich auch kaufen. Schnell ist man da viel Geld los. Doch taugen auch preiswertere Exemplare? Eine exemplarische Antwort am Beispiel eines Billig-Modells aus dem Prospekt eines Lebensmittel-Discounters.



Bild 1. Beute im Zwinger: Nach der Jagd auf den E-Scooter hatte ich noch Zeit für ein paar Lebensmittel.

ELEKTROKLEINSTFAHRZEUGE

Seit 2019 darf man auch in D die kleinen Elektro-Tretroller wirklich und tatsächlich nutzen. Aber natürlich nicht ohne Auflagen:

- › maximal 20 km/h
- › nicht auf Gehwegen
- › frei ab 14 Jahren
- › keine Helmpflicht
- › abnehmbare „lichttechnische Einrichtungen“
- › Versicherungspflicht
- › Allgemeine Betriebserlaubnis
- › zwei unabhängige Bremsen
- › keine Blinker erforderlich
- › eine „helle Glocke“ nötig

Letztes Jahr war ich zum ersten Mal mit einem Leihwagen in der Innenstadt von Tel-Aviv unterwegs. An sich bin ich nicht zimperlich und passe mich recht schnell an die regionalen Fahrgepflogenheiten an. Andere Länder – anderer Fahrstil. Aber das hatte ich noch nicht erlebt: In dieser quirligen Stadt im Nahen Osten gibt es nicht nur enge Straßen und kaum Parkplätze, sondern man wird im dichtesten Verkehr links und rechts gleichzeitig von diversen Elektroflitzern überholt und bedrängt. Auch an der Strandpromenade stolpert man überall über die achtlos liegengelassenen Miet-E-Scooter. Ergo: ich wollte auch einen...

Schwenk nach Europa

In allen europäischen Ländern gelten unterschiedliche Regeln bezüglich dieser kleinen Scooter. Während in vielen europäischen Großstädten das Surren der kleinen Flitzer längst zum Alltag gehört (und sich die Fußgänger an diese neue „Gefahr“

gewöhnt haben), ist das in Deutschland Neuland, denn erst 2019 wurden sie zugelassen. Natürlich nicht ohne entsprechende bürokratische Regularien (siehe **Kasten Elektrokleinstfahrzeuge**).

Und wie es sich für das Land des „demnächst“ eröffnenden Hauptstadtflughafens BER gehört, machen diese Sonderregeln die Roller teurer, denn erst einmal muss sich die chinesische Produktion auf die deutsche Extrawurst einstellen. Das ist mittlerweile der Fall, und so haben die beiden wohl größten Lebensmittel-discounter Aldi und Lidl doch kurz vor Weihnachten 2019 entsprechende Modelle zum kleinen Preis im Angebot. Lidl machte den Anfang und hatte Ende November das Modell „Doc Green E-Scooter ESA 5000“ für unglaubliche 299 € im Programm. Aldi folgte kurz später mit einem Kampfprijs von 279 €.

Interessierte waren dank Internet schon vorab informiert. Deshalb wusste ich, dass



Bild 2. Der Inhalt des Kartons – der zusammengefaltete E-Scooter wartet auf die Lenker-Montage.

das Aldi-Modell mit einer Ladung nur halb so weit wie sein Lidl-Konkurrent kommt, der immerhin 22 km schaffen soll. Außerdem wusste ich auch, dass es mit dem Modell M365 ein praktisch baugleiches, aber teureres und nicht in D zugelassenes Gerät von Xiaomi [1] gibt, und diese chinesische Firma steht zumindest bei Smartphones für viel Leistung und Qualität bei niedrigen Preisen. Auch wenn Lidl diese Verbindung dementiert hat, war mir das den Mehrpreis wert, und deshalb stand ich am Verkaufstag schon zehn vor sieben im Dunkeln vor der örtlichen Lidl-Filiale. Und ich war nicht der Erste! Vor mir standen schon zwei Herren mit ähnlichem Baujahr, jeweils mit einem großen Einkaufswagen bewaffnet und startbereit zur Jagd durch die Gänge und zum Einfangen des begehrten Stücks. Bis 7:00 Uhr waren weitere acht Menschen hinter mir. Alle warteten schon fast britisch-höflich ruhig in der Schlange. Und die Tür ging auf und der Wettlauf begann. Tatsächlich hatten die Jäger auf den Startpositionen 1 und 2 auch die ersten beiden Scooter eingeladen. Und ich bekam den Dritten – von sechs Stück. Als ich wieder an der Kasse war, zeigte ein Blick zurück, dass alle weg waren. Uff – Glück gehabt! In **Bild 1** können Sie meine Jagd-Trophäe bewundern.

Erste Eindrücke

Zuhause angekommen habe ich das Paket in mein Elektronik-Labor im Keller verfrachtet und konnte natürlich nicht widerstehen: Es musste ausgepackt werden. Nachdem es rausgehoben und von den Schaumstoff-Formteilen befreit war, bot sich der Anblick von **Bild 2**. Die Lenksäule kann weggeklappt werden und der Lenker war noch mit vier Inbus-Schrauben zu befestigen. Auch ein Ladegerät und Papier – insbesondere die „Datenbestätigung“ – war dabei, wie **Bild 3** zeigt. Letztere ist wichtig, da sie einer „Allgemeinen Betriebserlaubnis“ entspricht und man das Gefährt nur mit ihrer Vorlage versichert bekommt. Komplett zusammengesraubt stand der Roller fünf Minuten später in ganzer Pracht vor mir (**Bild 4**).

Vor der Inbetriebnahme aber musste erst einmal klar sein, wie und wo man das Ding einschaltet und was die LEDs etc. bedeuten. Laut Handbuch drückt man zum Einschalten den schwarzen Knopf am Lenker. Dann leuchtet das sehr helle, zweistellige Display mit der korrekten Geschwindigkeit von 00 km/h auf und die LEDs zeigen den



Bild 3. Das „Zubehör“ umfasst das Ladegerät, ein paar Schrauben samt Inbus-Schlüssel und Papiere.



Bild 5. So sieht der Lenker samt „Armaturenbrett“ eingeschaltet aus. Der Akku ist leer.

Ladezustand des Akkus an. **Bild 5** zeigt den Lenker in Fahrposition. An dieser Stelle wurde mein Tatendrang jäh eingebremst, denn es war nicht viel Saft im Akku, und die Anleitung fordert ein erstes volles Aufladen. Das rote Paddle links ist die Rekupeationsbremse und das Blaue rechts ist eine Art Daumengas. Die Lenkergriffe sind nicht drehbar. Oben rechts sieht man den Bremshebel für die hintere Scheibenbremse.

Fünf Stunden später war der Akku voll. Es hätte jetzt losgehen können mit der ersten Testfahrt. Testweise im Keller ausprobiert: Ja, es fährt. Schon jetzt merkte ich: So viel Power kommt da nicht, auch wenn es sicherheitshalber noch der Eco-Modus war. Vor dem Loslegen wollte ich aber wissen, wie gut dieser E-Roller gemacht ist. Ein Blick auf den Rahmen zeigt, dass die Alurohre sauber verschweißt sind (**Bild 6**). Auch der Mechanismus zum Arretieren des Lenkers sieht sehr stabil aus. **Bild 7** zeigt das Vorderrad. Lidl spricht hier von „8,5“ Luftkammerreifen (Honeycomb)“ – allein



Bild 4. Komplett zusammengeschaubarer E-Scooter. Das war einfach.



Bild 6. Stabile Schweißnähte und Klapp-Mechanismus.

EIGENSCHAFTEN

- › Straßenzulassung
- › Reichweite: ca. 22 km
- › Geschwindigkeit: max. 20 km/h
- › Motor: Nennleistung 350 Watt
- › Li-Ion-Akku mit 36 V / 7,2 Ah (259 Wh)
- › Ladezeit: bis 7 h
- › Ladegerät: 42 V / 1,5 A
- › Pannensichere 8,5“ Luftkammerreifen (Honeycomb)
- › Rutschfeste Gummiauflage auf Trittbrett
- › 2 unabhängige Bremsen
- › Frontlicht: 20 Lux, weiße Reflektoren
- › Rücklicht: 10 Lux (Bremslicht), rote Reflektoren
- › Seitlich: Reflektoren in gelb
- › Klingel (vormontiert am Lenker)
- › Seitenständer
- › LED-Display für Geschwindigkeit und Ladezustand



Bild 7. Vorderrad mit Nabenmotor und Vollgummireifen.



Bild 8. Hinterrad mit Scheibenbremse und Arretierung für den niedergeklappten Lenker.

mir fehlt der Glaube. Mich erinnert das eher an ganz normale Vollgummireifen. Auf jeden Fall muss man keinen Luftverlust befürchten. **Bild 8** zeigt das Hinterrad mit überdimensionierter, gelochter Bremsscheibe, Heckbeleuchtung und Nummernschildhalter. Der graue Gummipfropfen oben dient zum Arretieren des abgeklappten Lenkers. So kann man den Scooter trotz seines Gewichts von immerhin 15 kg bequem tragen. Im **Kasten Eigenschaften** finden Sie weitere Angaben.

Innenleben

Für Elektroniker besonders interessant ist das, was im Gerät drin steckt. Also musste es gleich aufgeschraubt werden. Das ist einfach, denn unten befindet sich lediglich ein dickerer Kunststoffdeckel (**Bild 9**). 13 Schrauben später hat man ihn in der Hand. **Bild 10** zeigt, dass er nur mit einem Schaumstoffband abgedichtet ist. Alles kein großes Problem, denn bei eBay

gibt es solche Abdeckungen aus Alu für wenig Geld in verschiedenen Farben für den Xiaomi M365 (Suche nach „Bottom Battery Cover Xiaomi M365“) und die dürften passen. Ganz allgemein gibt es dort viel passendes Zubehör unter dem Stichwort „M365“.

Bild 11 zeigt das Wichtigste: den Akku. Er ist aus Rundzellen aufgebaut (10s3p) und sieht ganz ordentlich aus. Auch die Prospektangaben stimmen. Der Akku ist von unten an den Rahmen geschraubt und bei Defekten gut zu wechseln. Das Zweitwichtigste ist die Elektronik:

Bild 12 zeigt, dass sie noch einmal extra durch ein transparentes Kunststoffgehäuse vor Feuchtigkeit geschützt ist. Die sechs MOSFETs für eine dreiphasige Brückenschaltung sind gut zu sehen. Aus diesem Sachverhalt ergibt sich, dass es sich um einen elektronisch gesteuerten BLDC-Motor handelt, was die Sache effizien-

ent und langlebig macht. **Bild 13** zeigt die Ladebuchse, die mit einem Gummipfropfen geschützt ist. Auch hier imponieren wieder die schönen Schweißnähte.

Testfahrt

Nachdem alles wieder zusammengeschaubt war, ging es raus auf die Straße. Im YouTube-Video [2] können sie meine ersten, noch zaghaften Fahrversuche bewundern. Ich traute mich zuerst nur im Eco-Modus, der bis 15 km/h geht. Zu Recht!

Damit der Scooter Gas annimmt, muss man ihn zuerst durch Anschieben auf >2 km/h bringen. Das ist kein Problem, ein Schubs genügt. Diese kleine Unbequemlichkeit hat den Vorteil, dass der hohe Stromverbrauch beim Anfahren reduziert wird und die Akkus so länger halten. Los ging's und die 15 km/h fühlten sich nicht unsicher an, trotz der kleinen Rädchen. Aber dann kam das Umkehren, und dazu muss man ja bekanntlich abbremsen. Vorab: Ich habe mir nichts dabei gebrochen...

Selbstverständlich testete ich zuerst den linken roten Hebel mit der Rekuperationsbremse. Sie hat leider ein eher digitales Verhalten und der Scooter bremst abrupt und stark. Da ich darauf nicht vorbereitet war, bin ich fast über den Lenker geflogen. Also nochmal beschleunigt und dann mit der Hinterrad-Bremse entschleunigt. Auch da hat man kaum Gefühl im Hebel. Die Scheibenbremse ist viel zu heftig für meinen Geschmack. Würde da Laub liegen, wäre das Hinterrad weg. So weit, so heftig. Von Beschleunigung ist im Eco-Modus kaum etwas zu spüren. 350 W hätte ich mir anders vorgestellt. Aber vielleicht kommt die volle Leistung ja erst im Normal-Modus, dachte ich. Also angehalten und mit Doppeldruck auf dem schwarzen Taster umgeschaltet und ab ging's. Es mag ja



Bild 9. Der Unterboden ist aus Kunststoff und abschraubbar.



Bild 10. Abgeschraubter Deckel mit Schaumstoffdichtung.

sein, dass da irgendein messbarer Unterschied zwischen den Modi außer der Endgeschwindigkeit ist, aber ich konnte den kaum spüren. Immerhin fährt der Lidl-Scooter wie versprochen seine 20 km/h. Das muss genügen.

Fazit

Generell war ich positiv überrascht von der Qualität des Scooters. Er ist wirklich ordentlich und stabil gemacht. Auch der Akku enttäuscht nicht und man kann ihm die 22 km Reichweite voll zutrauen – ausprobiert habe ich es aber nicht, da ich nicht gerne schiebe. Die Elektronik sieht ebenfalls gut aus und die Beleuchtung ist hell und ausreichend. Dass man so einen Scooter für den Preis machen kann, erstaunt mich.

Nicht so gut hingegen fand ich die Beschleunigung. Das sollen 350 W sein, wirklich? Es kommt kein richtiges Elektrofeeling auf – außer beim Rekuperieren. Damit zum größten Manko: Die Bremsen sind einfach viel zu heftig. Während die Scheibenbremse sicher mit der Zeit etwas geschmeidiger wird, ist die Rekuperation einfach zu giftig. Wer das konzipiert hat, der hat nur einen Teil seines geistigen Potentials genutzt.

Alles in Allem würde ich trotz der gewöhnungsbedürftigen Bremsen zu einem positiven Gesamturteil kommen. Das Preis/Leistungs-Verhältnis ist schon sehr gut. Trotzdem würde ich Ihnen zwingend zu einer Testfahrt raten, wenn Sie sich oder Ihren Sprösslingen so einen E-Scooter kaufen wollen.

Noch ein Schmanke! am Rande: Ein guter Freund hat übrigens ausprobiert, ob dieser Scooter in den Frunk (Kunstwort aus Front und Trunk) seines nagelneuen Tesla Model X passt. **Bild 14** beweist, dass das klappt. Martin möchte damit in der Stadt herumfahren, falls sein Tesla an einer Ladesäule eine Pause einlegt. Eine gute Idee und der Scooter ist sicherlich billiger als ein Außenspiegel seines neuen Elektromobils. Aber das ist ja auch 12,4 Mal schneller und bietet die 885-fache Leistung. ◀

191251-01



Bild 11. Der Akku. Sieht ganz ordentlich aus.



Bild 12. In der verkapselten Elektronik sind die sechs MOSFETs gut zu sehen.



Bild 13. Ladebuchse auf der linken Seite des Scooters.



Bild 14. Scooter im Frunk eines Tesla Modell X (Bild: Martin Jepkens [3]).

WEBLINKS

- [1] **Xiaomi Scooter:** www.mi.com/global/mi-electric-scooter
- [2] **YouTube-Video der Testfahrt:** www.youtube.com/watch?v=9UNWDpAYaO4
- [3] **Martin Jepkens:** www.me-marl.de

Ultraschall- Rückwärtseinparkhilfe mit Arduino Uno

Von **Dogan Ibrahim** (Großbritannien)

In diesem Projekt wird das Ultraschall-Sensormodul KY-050 zusammen mit einem Arduino und einem aktiven Summer KY-012 verwendet, um dem Fahrer beim Rückwärts-Einparken des Fahrzeugs zu helfen. Je geringer der Abstand zu den Objekten wird, desto „höher“ ertönt der Summer und weist den Fahrer darauf hin, dass sich das Fahrzeug Objekten im Heck nähert. Natürlich genügt die Schaltung keinen professionellen Ansprüchen, aber sie zeigt, wie leicht man das Ultraschall-Sensormodul in einem Projekt zur Abstandsmessung einsetzen kann.

Das Modul KY-050 besitzt vier Anschlüsse zwischen Sender und Empfänger, wie man in **Bild 1** gut erkennen kann. Das Modul ist mit einer Ultraschall-Sender/Empfänger-Kombination mit der Bezeichnung HC-SR04 bestückt und weist die folgenden Merkmale auf:

- › Betriebsspannung: 5 V
- › Betriebsstrom: 2 mA
- › Erfasster Abstand: 2...450 cm
- › Eingangs-Triggersignal: 10 μ s TTL
- › Sensorwinkel: <15 Grad

Die Anschlüsse des Moduls KY-050 sind: **Vcc** Plus der Versorgungsspannung, **Trig** Triggereingang, **Echo** Empfangsausgang und **Gnd** Masse der Versorgungsspannung. Eine Betriebssequenz des Ultraschall-Sensormoduls KY-050 verläuft wie in **Bild 2** dargestellt:

- › ein 10 μ s langer Triggerimpuls wird vom Steuerprogramm an das Modul gesendet
- › das Modul sendet daraufhin acht 40-kHz-Rechteckimpulse aus und setzt den Echo-Pin auf High

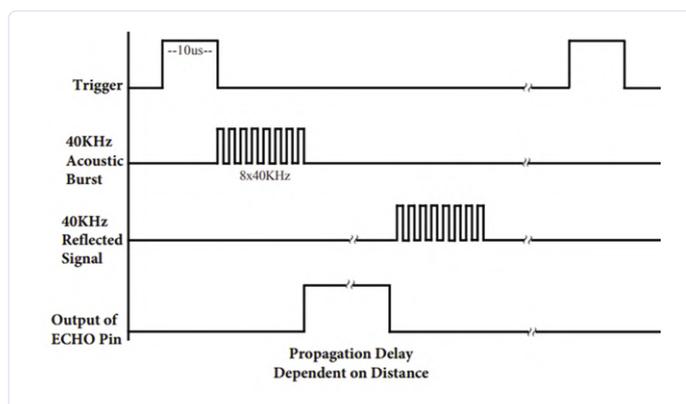


Bild 2. Funktionsweise des Ultraschall-Sensormoduls.

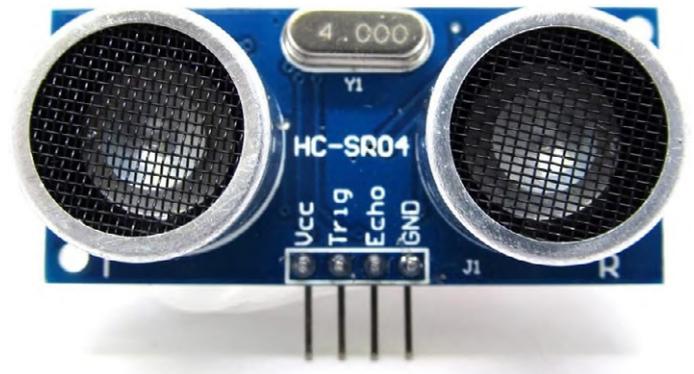


Bild 1. Das Ultraschallmodul KY-050.

- › das Programm startet einen Timer
- › das Signal trifft auf das Ziel und kehrt als Echo zum Modul zurück
- › wenn das rückkehrende Echo vom Modul erkannt wird, geht der Echo-Pin auf Low
- › das Programm erhält diese Information und stoppt den Timer
- › die Dauer des Echosignals, die proportional zur Entfernung zum Ziel ist, wird vom Programm in eine Entfernung umgerechnet.

Diese Umrechnung im Programm ist nicht besonders kompliziert.

Entfernung zum Objekt (in m) =

$$\text{Dauer der Echozeit (in Sekunden)} \times \text{Schallgeschwindigkeit} / 2$$

Die Schallgeschwindigkeit beträgt ungefähr 340 m/s oder 0,034 cm/ μ s, woraus sich ergibt:

Entfernung zum Objekt [cm] =

$$(\text{Dauer der Echozeit } [\mu\text{s}] \times 0,034) / 2$$

beziehungsweise

$$\text{Entfernung zum Objekt [cm]} = (\text{Dauer der Echozeit } [\mu\text{s}]) \times 0,017$$

Wenn also beispielsweise das Echosignal 294 μ s high ist, wird der Abstand zum Objekt so berechnet:

$$\text{Entfernung zum Objekt [cm]} = 294 \mu\text{s} \times 0,017 \text{ cm}/\mu\text{s} = 5 \text{ cm}$$

Bild 3 zeigt so etwas wie ein Blockdiagramm und **Bild 4** das Schalt-

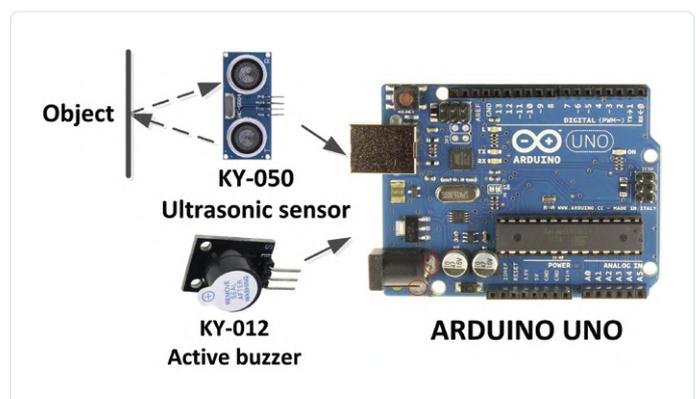


Bild 3. Blockschaltung des Projekts im Embedded-Style.

Listing 1. Das Programm „ultrasonic“ für Arduino.

```

/*****
 *   ULTRASONIC REVERSE PARKING WITH BUZZER
 *   =====
 *****/
int trig = 2;           // trig pin
int echo = 3;          // echo pin
int buzzer = 4;        // buzzer
int dely;

long tim;
float distance;

void setup ()
{
  Serial.begin(9600);
  pinMode(trig, OUTPUT);    // trig is output
  pinMode(echo, INPUT);    // echo is input
  pinMode(buzzer, OUTPUT); // buzzer is output
  digitalWrite(buzzer, LOW); // buzzer OFF
}

void loop ()
{
  digitalWrite(trig, LOW); // clear trig
  delayMicroseconds(5);   // 5us delay

  digitalWrite(trig, HIGH); // set trigger HIGH
                          // for 10us
  delayMicroseconds(10);  // 10us delay
  digitalWrite(trig, LOW); // remove trigger

```

```

tim = pulseIn(echo, HIGH); // read the echo
distance = tim * 0.034 / 2; // calculate distance
Serial.println(distance);  // display distance

//
// Set the delay depending on the distance
// to the object
//
if(distance > 100)
  dely = 0;
else if(distance > 70 && distance < 90)
  dely = 600;
else if(distance > 50 && distance < 70)
  dely = 400;
else if(distance > 30 && distance < 50)
  dely = 300;
else if(distance > 10 && distance < 30)
  dely = 200;
else if(distance < 10)
  dely = 10;

if(distance < 100) // if less than 100cm
{
  digitalWrite(buzzer, HIGH); // buzzer ON
  delay(dely);                // delay
  digitalWrite(buzzer, LOW);  // buzzer OFF
  delay(dely);                // delay
}
}

```

bild, so wie Sie es gewohnt sind. Die folgenden Verbindungen werden zwischen KY-050, KY-012 und Arduino Uno gelegt:

Arduino Uno	KY-050	KY-012
2	trig	
3	echo	
4	S	
GND	GND	GND
+5V	Vcc	

Das Summer-Modul KY-012 wird direkt an den Arduino-Uno-Port-Pin 4 angeschlossen. Es kann nicht schaden, einen Reihenwiderstand von 100 Ω in den Modulanschluss aufzunehmen, um den Port-Ausgangsstrom zu begrenzen, insbesondere wenn auch andere Bauteile Strom von den Port-Pins ziehen.

Im Listing 1 werden zu Beginn Trigger, Echo und Summer den Portnummern 2, 3 und 4 zugeordnet. Innerhalb der Setup-Routine

wird der Summer als Ausgang konfiguriert und zunächst ausgeschaltet. Außerdem werden Trig als Ausgang und Echo als Eingang konfiguriert. Innerhalb der Hauptprogrammschleife wird ein Triggerimpuls gesendet und der Befehl `pulseIN` verwendet, um das Echosignal zu lesen. Der Abstand zwischen Hindernis und Ultraschallsensor wird dann (in Zentimetern) berechnet und in einer Variablen namens `distance` gespeichert. Der berechnete Abstand wird in der Variablen `dely` (kein Tippfehler!) gespeichert. Je kleiner der Anstand zum Objekt, desto niedriger ist auch der Wert von `dely`. Schließlich wird der Summer im Takt der Variablen `dely` aktiviert. Das Ergebnis ist, dass die Frequenz umso höher wird, je näher der Sensor dem Objekt kommt. Doch sollte es nur darum gehen, hätten wir uns die ganze Umrechnung in Zentimeter auch sparen können. Allerdings lässt sich auf diese Weise die Entfernung auf dem seriellen Monitor (oder einem Display?) im „Klartext“ anzeigen. Zum Abschluss zeigt Bild 5 die auf einer Lochrasterplatine aufgebaute funktionsfähige Schaltung. Nicht schlecht für einen Sonntagnachmittag! ◀

200211-03

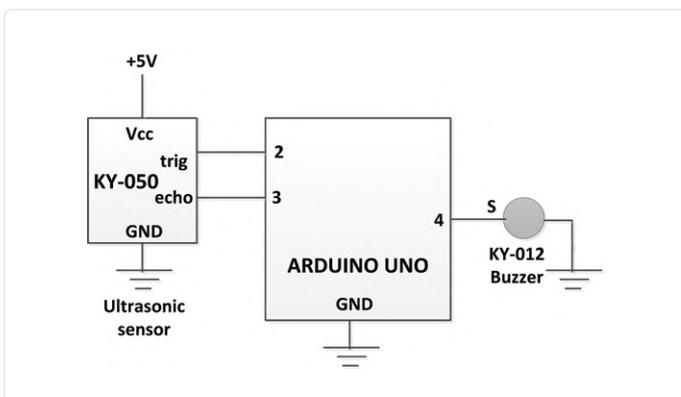


Bild 4. Schaltplan des Projekts.

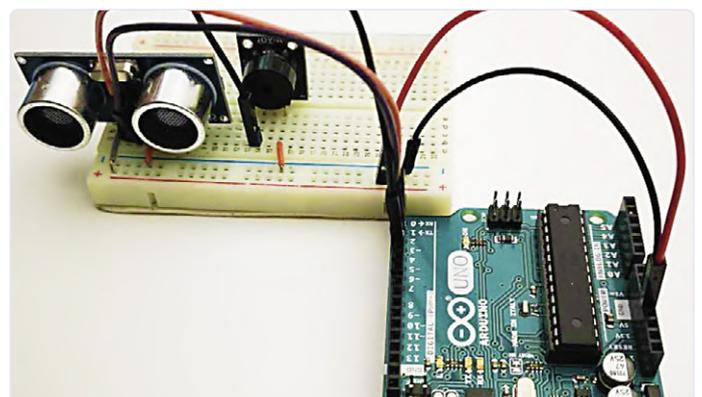


Bild 5. Die auf einer Lochrasterplatine aufgebaute Schaltung.

Verzerrer-Pedal mit Operationsverstärker und Raumladungsrohren



Von **Richard Honeycutt**

Bei diesem Projekt handelt es sich um eine „hybride“ Elektronik, die integrierte Schaltungen und Röhren kombiniert. Es wurde vom Autor als Demonstrationsaufbau entwickelt, der in seinem im Elektor-Verlag erschienenen Buch *The State of Hollow State Audio - in the Second Decade of the 21st Century* beschrieben wird. Die verwendeten 12EL6 sind Niederspannungs-Raumladungsrohren, die in dem Buch ausführlich besprochen werden.

Bei dem im Schaltplan in **Bild 1** gezeigten IC handelt es sich um einen Operationsverstärker der Typs OP27. Der ursprünglich verwendete LTC6090 erwies sich als in Kleinstmengen zu schwer erhältlich. Da der Eingangswiderstand des Opamps etwa 1 M Ω betragen muss, sollten Bias-Strom, Offset-Strom und Offset-Spannung sehr niedrige Werte aufweisen. Außerdem muss der Operationsverstärker sehr rauscharm sein. Der OP27 erfüllt all diese Anforderungen und ist in kleinen Stückzahlen leicht erhältlich. Der LTC6090 verfügt allerdings über Rail-to-Rail-Ein- und Ausgänge, während beim OP27 bei einer Last von 2 k Ω die maximale Ausgangsspannung typischerweise 1,5 V (maximal 3,5 V) unter der Versorgungsspannung liegen muss. Dies muss bei der Einstellung berücksichtigt werden und erfordert Maßnah-

men, damit der Operationsverstärker zur Einstellung der Steuergittervorspannung von V1 verwendet werden kann. Es gibt zwar auch andere Low-Power- und Rail-to-Rail-Operationsverstärker, aber ich habe keinen gefunden, der gut in Einzelstückzahlen erhältlich ist, so dass es beim OP27 bleiben soll.

Die Ausgangsgleichspannung des Opamps muss (im Ruhezustand) mindestens 3,5 V positiver sein als die negative Versorgung, die ja bei einer einfachen, asymmetrischen Spannungsversorgung gleichbedeutend mit Masse ist. Wir müssen auch etwa 1 V Reserve für den Ausgangsspannungshub einkalkulieren, um ein Clipping des Opamps zu vermeiden. Und dann muss die Steuergittervorspannung von V1 im Bereich 0...-0,2 V einstellbar sein. Natürlich können wir mit einer einfachen Versorgung nicht so einfach eine negative Spannung erzeugen, also müssen wir statt dessen die Kathode etwas positiver machen. In der ursprünglichen Schaltung wurden zwei Dioden in Reihe zwischen Kathode und Masse geschaltet, um die Kathode etwa 1,2 V über Masse anzuheben. Bei einer Steuergitterspannung von 1 V erhalten wir dadurch so die gewünschte Vorspannung von -0,2 V. Aber wir können den Ausgang des OP27 im Ruhezustand nicht so niedrig aussteuern. Da der Spannungsabfall im Durchlassbereich einer Halbleiterdiode nicht genau definiert ist (ungefähr 0,6 V), musste ich experimentell herausfinden, dass ich den gewünschten Bereich der Steuergittervorspannung erhalten kann, wenn ich nur eine Diode zwischen Kathode und Masse von V1 verwende und den nicht-invertierenden Eingang des Opamps in einem Bereich von 0,387 V bis 5,6 V einstellen kann.

Da ich aus einer früheren Arbeit wusste, dass der Eingangswiderstand des Kathodenfolgers die Stufe mit V1 belasten würde, untersuchte ich das Ausgangssignal von V1. Ich stellte fest, dass ein stärkeres Signal erforderlich war, als ein typischer Humbucker erzeugen würde, um die von mir gewünschte „distortion signature“ zu erzeugen. Also erhöhte

Grundlegendes für den Elektronik-Arbeitsplatz

Vom **Elektor Team**

Ob Ingenieur oder Maker – alle brauchen einen vernünftig ausgestatteten Elektronik-Arbeitsplatz für die Entwicklung, den Bau und den Test Ihrer Projekte. Selbstverständlich gibt es auch hier durch ein realistisches Budget vorgegebene Grenzen.

Man kann nicht alles einfach anschaffen, was interessant oder toll ist. Wenn Sie nach Tipps, Hinweisen und Vorschlägen suchen, sind Sie bei der folgenden grundlegenden Zusammenfassung richtig.

Elektronik-Arbeitsplatz: Möbel & Beleuchtung

Zuallererst benötigen Sie eine stabile, flache Oberfläche – vorzugsweise in weißer Farbe. Dies ermöglicht eine bessere Ausleuchtung



durch erhöhte Reflexion und hilft auch, (winzige) Bauteile und Werkzeuge schnell zu finden und zu erkennen.

Niemand will einen schmerzenden Rücken. Holen Sie sich einen hochwertigen, höhenverstellbaren Drehstuhl, um längere Löt- und Designsitzungen komfortabel zu gestalten.

Achten Sie außerdem auf viel Licht. Eine gute LED-Lampe sollte für Rechtshänder in einer linken oberen Seitenposition auf Ihrer Werkbank positioniert werden und für Linkshänder umgekehrt. Auf diese Weise fällt kein Schatten von Ihrer Arbeitshand auf das Werkstück.

Sie werden wahrscheinlich ganz nah an Ihrem Projekt dran sein wollen, denn moderne Bauteile werden immer kleiner und kleiner. Eine Lösung wie die LED-Lupe Toolcraft Helping Hand (**Bild 1**) hilft, indem sie eine exzellente LED-Beleuchtung mit ausreichender Vergrößerung kombiniert und zudem eine exzellente „dritte“ / helfende Hand bietet.

Organisation des Elektronik-Arbeitsplatzes

Die Organisation Ihres Elektronik-Arbeitsplatzes sollte sowohl individuell als auch ergonomisch sein. Wenn Ihre Ausrüstung nur eine Armlänge entfernt ist, macht das das Elektronikerleben weniger stressig, effektiver und sinnvoller. Dann macht Elektronik mehr Spaß.

Bei verwirrten Kabeln und Drähten neigt jeder Elektroniker zum Fluchen. Dieses Problem kann durch Nutzung einfacher Drahtrollen gelöst werden. Lächerlich einfach zu bauen: Man braucht nur etwas Metall zwischen zwei quadratischen Holzstücken.

Hier einige weitere einfache Tipps, um die Effektivität und Ordnung zu verbessern:

- Verwenden Sie beschriftete Marmeladengläser für Ersatzteile, Befestigungsmittel, Schrauben usw.
- Verwendung einer Farbkennzeichnung für verschiedene Bauteilarten



Bild 1. Toolcraft Helping Hand mit LED-Lupe.



- › Erwägen Sie den Kauf von vorsortierten Werkzeug-Kits (**Bild 2**). iFixit [1] ist z. B. spezialisiert auf hochwertige, leicht zugängliche Kits in praktischen Beuteln mit allen benötigten Werkzeugen (kein Unsinn, den Sie nicht benötigen!).

Prüf- und Messgeräte

Wenn es um die Überprüfung von Geräten geht, taugt nur Qualität. Qualität ist in der Elektronik gleichbedeutend mit Genauigkeit. Hier sind einige Tipps und Tricks, die Ihnen den Einstieg erleichtern:

- › **Multimeter:** Ein gutes Multimeter ist das wichtigste Tool jedes Elektrikers. Sie benötigen dies, um Durchgang, Widerstand, Kapazität, Strom und Spannung zu messen. Darüber hinaus sollte es auch über einen Mikroampere-Bereich verfügen. Um sicher zu sein, dass Ihr Multimeter gut genug ist, lohnen sich bekannte Marken wie PeakTech, Siglent, Fluke, Voltcraft

usw. Elektor hat eine breite Palette von hochwertigen Multimetern [2] im Programm.

- › **Stromversorgung:** Sie benötigen mindestens ein Labornetzteil, vorzugsweise eines mit variabler Ausgangsspannung und einstellbarem Maximalstrom. Man kann nie genug Netzteile haben. Mit der Zeit hat man dann zwei, drei oder gar vier davon.

- › **Signalgenerator:** Es ist immer irgendwo nötig, ein Signal zu erzeugen. Daher ist ein Signalgenerator mit Grundsignalen wie Sinus, Dreieck und Rechteck ein Muss.
- › **Oszilloskop:** Bei auch nur ein bisschen anspruchsvolleren Projekten kommt man um ein Oszilloskop nicht herum, denn eine visuelle Inspektion von dynamischen Signalen ist sehr hilfreich. Stand-alone-Geräte können manchmal sperrig und umständlich sein. Hier bieten sich auch kleine und moderne USB-Alternativen wie das SmartScope [3] an, siehe **Bild 3**.

Werkzeuge, Werkzeuge, Werkzeuge...

„Ein Handwerker ist nur so gut wie sein Werkzeug.“ Dieses alte Sprichwort gilt nach wie vor. Qualitätswerkzeuge sind die Basis von Qualitätsarbeit. Möchten Sie wissen, welche Tools besonders relevant sind? Hier einige Empfehlungen:

- › **Löttechnik:** Das grundlegendste Werkzeug jedes Elektrikers ist? Sie haben es erraten: Der LötKolben. Heute in Form einer einstellbaren Lötstation. Ob Sie Reparaturen durchführen, Schaltungen entwickeln oder „nur basteln“ – ein hochwertiges Lötgerät ist entscheidend für das optimale Ergebnis. Man muss nicht unbedingt gleich ein paar hundert Euro ausgeben. Aber ein anständiges Markengerät mit einstellbarer Temperatur und nicht zu großer Stellfläche sollte es schon sein. Ein Beispiel: Das Modell Weller WT 1014 [4] ist zwar nicht sonderlich preiswert, aber eben auch nicht „billig“, sondern bietet eine ausgezeichnete Komplettlösung. Denken Sie auch daran eine gute Entlötpumpe und eine lötfeste Unterlage zu kaufen. Damit wird die Arbeitsplatte auch auf längere Sicht sauber und ansehnlich bleiben.
- › **Lüfter:** Anstatt viel Geld für eine Lötdampfabsaugung auszugeben, hilft ein einfacher, normaler Desktop-Lüfter, die bösen Lötdämpfe loszuwerden. Außerdem sorgt er in den heißen Sommermonaten für einen kühlen Kopf.



Bild 2. iFixit toolKit.



Bild 3. SmartScope USB-Oszilloskop.

- › **Seitenschneider und Zangen:** Zwei unterschiedlich große, aber präzise und bündig schneidende Seitenschneider sind absolut notwendig, ebenso wie Spitz- und Flachzangen. Die Marke Lindstrom bietet z. B. eine breite Palette solcher Werkzeuge.
- › **Kleber:** Der Heißkleber aus einer Klebepistole hat eine Vielzahl von Anwendungen: Zu den wichtigsten gehören die mechanische Fixierung elektronischer Bauteile und von Drähten sowie die elektrische Isolierung und sonstige Klebereien bei der Elektronikmontage.
- › **Abisolierzangen:** Da die Zweckentfremdung eines Seitenschneiders zu suboptimalen Ergebnissen führt, sind passende Abisolierzangen eine nützliche Ergänzung Ihrer Werkzeugsammlung.
- › **Messschieber:** (Digitale) Messschieber sind auch für nicht so feinmechanikaffine Elektroniker unersetzlich. Zusätzlich wird man vermutlich immer noch ein einfaches Metalllineal benötigen.
- › **Messer:** Ein gutes Messer oder Skalpell ist wichtig, um Leiterbahnen auf Platinen zu überarbeiten und für vieles andere mehr.
- › **Schraubendreher:** Elektroniker nehmen viel auseinander und bauen es wieder zusammen. Es braucht also gute Schraubendreher hoher Qualität. Statt wilder Sammlungen zweifelhafter Provenienz sollte man in ein paar Schraubendreher-Kits mit Schlitz-, Kreuz-, Torx- und Inbus-Spitzen investieren. So sorgen Sie für reibungsloses und stressfreies Arbeiten.

Besorgen Sie sich die wichtigsten Teile zuerst. Bei einem knappen Budget erstellen Sie eine Liste der am häufigsten verwendeten Komponenten und kaufen Sie zunächst einmal diese. Setzen Sie auf Klasse statt Masse.

Schlüsselkomponenten und Zubehör

Wir empfehlen die folgenden Schlüsselkomponenten und Zubehörteile.

- › **Lötfreies Steckbrett:** Die gängigste Technik für das Prototyping ist die Verwendung von lötfreien Steckbrettern. Diese ermöglichen einen stressfreien Test Ihrer Schaltungsideen und die unkomplizierte Änderung und Verbesserung.
- › **Streifenleiterplatinen:** Eine andere, aber beliebte Methode für das Prototyping Ihrer Schaltungen ist die Verwendung von Experimentierplatinen mit Streifenleitern oder Löt pads (die von Veroboard sind ziemlich beliebt).
- › **Elektronische Bauteile:** Es ist immer eine gute Idee, eine grundlegende Auswahl an Widerständen, Kondensatoren,

Induktivitäten, Dioden, Transistoren und LEDs zur Hand zu haben.

- › **Lötzinn:** Vermeiden Sie zu dicken Löt draht. Dünnes Lötzinn $\leq 0,5$ mm ist super und Maker können für nichtindustrielle Produkte immer noch das besser zu verarbeitende, eutektische Bleilöt 60/40 verwenden.
- › **Kältespray:** Unverzichtbar, um defekte ICs und kalte Lötstellen auf Platinen zu finden. Auch nützlich auch beim Lösen von Befestigungselementen und Schrauben.
- › **Druckluft(spray):** Ideal für die Reinigung, verhindert auch elektrostatische Aufladung im Zusammenhang mit anderen Reinigungsmethoden.
- › **Lösungsmittel:** Isopropyl und sonstige Lösungsmittel sind für eine Vielzahl von Anwendungen geeignet.
- › **Kleber:** Klebestifte für Ihre Klebepistole, Epoxidharz (Araldite, JB Weld) und Loctite, um Ihre Schrauben und Befestigungselemente an Ort und Stelle zu halten.
- › **Leitungen:** Besonders nützlich sind welche mit Kokodilklemmen und (Sicherheits-) Bananensteckern an den Enden.
- › **Sicherheitsausrüstung:** Für die Sicherheitsbewussteren unter uns sind Antistatik-Armbänder und vielleicht ein Erste-Hilfe-Set erforderlich.

Entwerfen, bauen und verkaufen von Elektronik

Es sollte keine Unsummen kosten, einen Elektronik-Arbeitsplatz einzurichten. Man kann es nur nochmals wiederholen: Die Qualität ist entscheidend. Die Zeit, die Sie mit der Recherche und der Berücksichtigung der hier vorgestellten Aspekte verbringen, spart Ihnen langfristig Zeit, Nerven und Geld. Als Ergebnis haben Sie einen richtig guten Arbeitsplatz vor sich. Hoffentlich hat Ihnen dieser Artikel einige Ideen geliefert, was Sie beim Kauf Ihrer Ausrüstung so brauchen.

Elektor konzentriert sich darauf, Ingenieuren, Makern und Studenten dabei zu helfen, Elektronik zu entwickeln, zu bauen und zu verkaufen. Melden Sie sich für einen kostenlosen Elektor-Labs-Account [5] an, um die in Ihrem neu überarbeiteten Arbeitsplatz entwickelten Projekte zu präsentieren und mit Entwicklern auf der ganzen Welt zusammenzuarbeiten!

Wenn Sie Ihren Arbeitsplatz fertig haben, schauen Sie auf der Webseite [6] vorbei, und senden Sie uns Fotos und ein paar Worte zu ihrem Labor! ◀

200231-03

WEBLINKS

- [1] **Toolkits:** www.elektor.de/tools/production/toolkits
- [2] **Multimeter:** www.elektor.de/tools/test-measurement/multimeters
- [3] **SmartScope:** www.elektor.de/labnation-smartscope-usb-oscilloscope
- [4] **Weller WT-1014:** www.elektormagazine.de/news/review-lotstation-wt1014-von-weller
- [5] **Elektor Labs Webseite:** www.elektormagazine.de/labs
- [6] **My Workspace Submission Form:** www.elektormagazine.com/workspace-submission

Hexadoku Sudoku für Elektroniker

Wie in jeder Ausgabe finden Sie auch in diesem Heft unser ganz spezielles Sudoku. PC, Oszilloskop und Lötkolben können sich erholen, während Ihre kleinen grauen Zellen auf Hochtouren arbeiten. Wenn Sie alle Hex-Ziffern in den grauen Kästchen herausgefunden haben, sollten Sie uns diese gleich zumailen – denn hier warten fünf Elektor-Gutscheine!

Die Regeln dieses Rätsels sind ganz einfach zu verstehen: Bei einem Hexadoku werden die Hexadezimalzahlen 0 bis F verwendet, was für Elektroniker und Programmierer ja durchaus passend ist. Füllen Sie das Diagramm mit seinen 16 x 16 Kästchen so aus, dass alle Hexadezimalzahlen von 0 bis F (also 0 bis 9 und A bis F) in jeder Reihe, jeder Spalte und in jedem

Fach mit 4 x 4 Kästchen (markiert durch die dickeren schwarzen Linien) **genau einmal** vorkommen. Einige Zahlen sind bereits eingetragen, was die Ausgangssituation des Rätsels bestimmt. Wer das Rätsel löst – sprich die Zahlen in den grauen Kästchen herausfindet – kann einen von fünf Gutscheinen im Wert von 50 Euro gewinnen!



EINSENDEN

Schicken Sie die Lösung (die Zahlen in den grauen Kästchen) per E-Mail, Fax oder Post an:

Elektor Redaktion
Kackertstr. 10
52072 Aachen

Fax: 0241 / 955 09-013

E-Mail: hexadoku@elektor.de

Als Betreff bitte nur die Ziffern der Lösung angeben!

Einsendeschluss ist der 3. August 2020.

Die Gewinner des Hexadokus aus der Ausgabe Mai/Juni 2020 stehen fest!

Die richtige Lösung ist: **638CB**

Einen Elektor-Wertgutschein über je 50 € haben gewonnen:

Ralf Kloos, Joseph Reding, Ola Sandin, Isolde Tietz und Morris Beavers.

Herzlichen Glückwunsch!

6	D		A			3	5	C	E			2	8	1		
F			5	4		C			A			2	6		3	
4	1													A	0	
3	B		8	F	0							5	6	7	D	4
0			1				3	E					8			2
	6															4
	5	3	4		6		8	B		2		0	7	9		
			7	9			A	0				4	B			
						C	1		E	D		8	3			
	3					D	4					E	1			6
							6	0	F	4						
	E			8	F				6	C						0
D	2	A		0	7				C		9		8	3	E	
		B												2		
9	8		E		C						0		A		5	F
5		F	3				6	4					1	0		C

9	B	C	E	6	5	1	4	0	7	F	2	3	A	D	8	
4	D	8	F	A	7	E	0	9	B	1	3	5	C	6	2	
0	A	3	5	F	2	9	D	8	C	4	6	1	7	E	B	
1	7	2	6	3	8	C	B	D	5	E	A	9	F	4	0	
2	3	A	C	1	9	8	F	7	6	5	0	4	D	B	E	
B	4	5	9	D	A	2	E	1	3	8	C	6	0	7	F	
D	0	E	7	B	3	4	6	2	A	9	F	8	1	C	5	
F	1	6	8	0	C	5	7	4	D	B	E	A	9	2	3	
C	5	9	0	4	6	B	8	A	E	7	D	2	3	F	1	
A	2	D	3	7	1	F	C	6	4	0	8	B	E	5	9	
E	6	1	4	9	D	A	3	B	F	2	5	7	8	0	C	
7	8	F	B	5	E	0	2	C	9	3	1	D	4	A	6	
3	9	0	2	C	F	7	A	5	8	D	B	E	6	1	4	
5	C	7	A	E	B	3	9	F	1	6	4	0	2	8	D	
6	F	4	1	8	0	D	5	E	2	A	9	C	B	3	7	
8	E	B	D	2	4	6	1	3	0	C	7	F	5	9	A	

Der Rechtsweg ist ausgeschlossen. Mitarbeiter der in der Unternehmensgruppe Elektor International Media B.V. zusammengeschlossenen Verlage und deren Angehörige sind von der Teilnahme ausgeschlossen.

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!

Unser Angebot des Monats



Elektor Raspberry Pi
Elektronik Kit

Preis: 129,95 €

Mitgliederpreis: 116,96 €

 www.elektor.de/19267



Picade Desktop Retro Arcade-
Automat für RPi (10" Display)

Preis: 259,00 €

Mitgliederpreis: 233,10 €

 www.elektor.de/19275

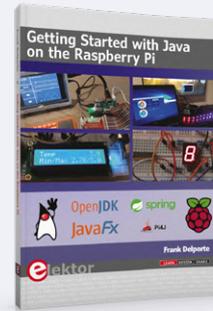


Raspberry Pi High Quality Kameramodul

Preis: 59,95 €

Mitgliederpreis: 53,95 €

www.elektor.de/19279



Getting Started with Java on the Raspberry Pi

Preis: 34,95 €

Mitgliederpreis: 31,46 €

www.elektor.de/19292



Qoitech Otii Arc – Netzteil, Power Meter und Datenerfassung

Preis: 639,00 €

Mitgliederpreis: 575,10 €

www.elektor.de/19270



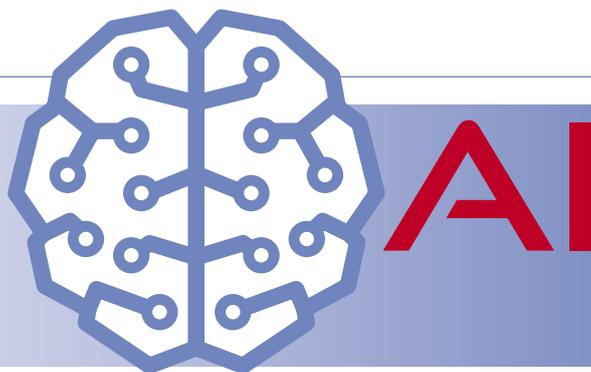
Ringo – Edukatives DIY-Mobiltelefon-Kit

Preis: 144,95 €

Mitgliederpreis: 130,46 €

www.elektor.de/19269

KI für Einsteiger (2)



Neuronale Netze mit Linux und Python

Von Walter Trojan

Im ersten Artikel dieser kleinen Serie [1] wurde die Hardware des Maixduino vorgestellt und gezeigt, wie man diesen mit der Arduino-IDE in C++ programmieren kann. Dabei wurde die Leistungsfähigkeit des Prozessors mit einem KI-Modell zur Objekterkennung demonstriert. Wie die künstliche Intelligenz beim vertieften Lernen funktioniert, soll in diesem Beitrag angerissen werden. Wer sich ernsthaft mit KI beschäftigen möchte, kommt zudem an Linux und Python nicht vorbei. Doch mit den entsprechenden Tools ist das kein Hexenwerk.

Bitte erwarten Sie nicht, dass Sie nach dem Lesen der nachfolgenden Passagen das Deep Learning vollständig verstanden haben. Dazu waren bei mir das Studium einiger Bücher und Tutorials sowie zahlreiche Programmierversuche erforderlich (und vieles weiß ich noch immer nicht). Aber ich möchte Ihnen zumindest einen Überblick vermitteln, welche Strukturen und Methoden eingesetzt werden, um ein universell einsetzbares Programm in Form eines Neuronalen Netzes (NN) mit einer spezialisierten Intelligenz auszustatten.

Aufbau eines Neuronalen Netzes

Ein Neuronales Netz (NN) besteht aus mehreren Schichten, die jeweils mehrere Knoten aufweisen. In **Bild 1** sind die Knoten einer Schicht vertikal angeordnet. Dabei können pro Schicht theoretisch beliebig viele Knoten und im Netz beliebig viele Schichten vorhanden sein. Die Architektur wird von der jeweiligen Aufgabe bestimmt und die Größe ist natürlich auch abhängig von den Ressourcen der benutzten Rechnerplattform. Soll das

NN z.B. Objekte klassifizieren, die ihm per Bild übermittelt werden, übernimmt die erste Schicht die Daten zur Lösung der Aufgabe und stellt zur Erfassung eine entsprechende Anzahl von Eingangsknoten zur Verfügung. Besteht ein niedrig aufgelöstes Bild nur aus $28 * 28$ Pixeln, dann sind bei einer Darstellung in Grauwerten 784 Knoten erforderlich. Zur Erfassung von Farbbildern würde sich die Anzahl verdreifachen. Bei der Bilderfassung erhält jeder Eingangsknoten den Grauwert „seines“ Pixels.

Das Ergebnis der Bildanalyse wird in der Ausgangsschicht präsentiert, die für jedes Resultat einen eigenen Knoten vorhält. Sollte das NN 1000 Objekte erkennen, hätte diese Schicht die gleiche Anzahl von Ausgangsknoten. Dabei zeigt jeder Knoten eine Wahrscheinlichkeit zwischen 0 und 1,0 an und signalisiert damit, wie sicher sich das NN beim gefundenen Resultat ist. So könnte bei der Aufnahme einer Hauskatze der zugehörige Knoten einen Wert von 0,85 und der Knoten „Tiger“ einen Wert von 0,1 aufweisen. Alle anderen Knoten hätten dann noch geringere Wahrscheinlichkeiten und das Ergebnis wäre damit eindeutig.

Die eigentliche Analysearbeit wird beim Deep-Learning von den verdeckten Schichten erledigt. Abhängig von der Aufgabe können beliebig viele verdeckte Schichten eingesetzt werden; in der Praxis sind durchaus 100 bis 200 solcher Schichten im Einsatz. Dabei kann man neben der oben gezeigte Architektur

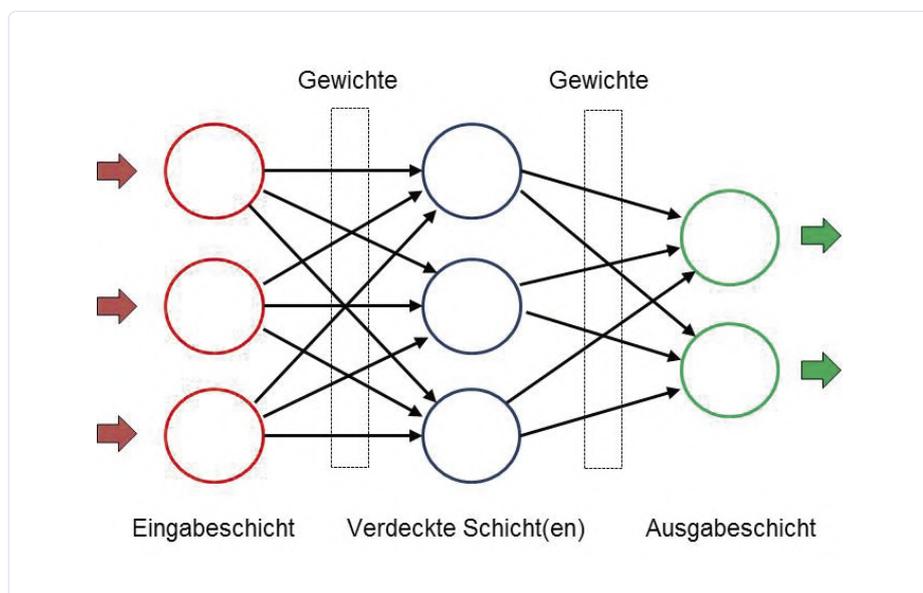


Bild 1. Struktur eines Neuronalen Netzes.

auch komplexere Strukturen mit Rückkopplungen oder zwischengeschalteten Filtern zur Steigerung der Genauigkeit finden. Die Anzahl der Knoten in jeder verdeckten Schicht ist ebenfalls beliebig. Im oben erwähnten Beispiel zur Objekterkennung könnten bereits drei verdeckte Schichten mit jeweils 200 Knoten schon brauchbare Resultate liefern.

Wird beim menschlichen Gehirn die Intelligenz durch die Kopplung der Neuronen mittels Synapsen erzielt, sind auch beim NN die Knoten miteinander vernetzt. Dabei ist jeder Knoten einer Schicht mit sämtlichen Knoten der nachfolgenden verbunden. Jede Verbindung (im Bild als Pfeil dargestellt) enthält einen Wert, der als Gewicht bezeichnet ist. Diese Gewichte zwischen den Schichten sind in Matrizen gespeichert. Deshalb werden im Bereich des Deep Learnings Programmiersprachen bevorzugt, bei denen Matrix-Operationen einfach und schnell durchgeführt werden können. Und wie erzielt nun ein NN die gewünschten Resultate? Jeder Knoten erhält seine Eingangssignale von allen Knoten der vor ihm liegenden Schicht, in **Bild 2** mit x bezeichnet. Diese Werte werden mit den Gewichtswerten w multipliziert und addiert, also ist die Netzeingabe

$$\text{net} = x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 \dots$$

Der errechnete Wert von net wird nun mit einer Aktivierungsfunktion multipliziert und dem Ausgang (und damit allen Knoten der Folgeschicht) zugeleitet. Ein knotenspezifischer Schwellwert bestimmt darüber hinaus, ob dieser Knoten „feuert“.

Die in **Bild 3** gezeigten Aktivierungsfunktionen werden den Schichten zugeordnet und sorgen dafür, dass die Ausgangswerte im gewünschten Bereich bleiben. So unterdrückt die Relu-Funktion alle negativen Werte und bei Sigmoid erfolgt eine Begrenzung zwischen 0 und 1. Durch diese Maßnahmen arbeiten NNs in einem übersichtlichen Zahlenbereich und können nicht durch „Ausreißer“ dominiert werden.

Training

Nach der Erfassung der Eingangsdaten führt das NN also in jeder Schicht zahlreiche Berechnungen durch und präsentiert die Ergebnisse am Ausgang. Dieser Vorgang wird mit Inferenz bezeichnet. Und wie kommt nun die Intelligenz dort hinein? Das erfolgt wie bei einem Kleinkind durch Schulung und Training. Bei einem untrainierten NN sind die Gewichte meistens mit Zufallszahlen im Wertebereich

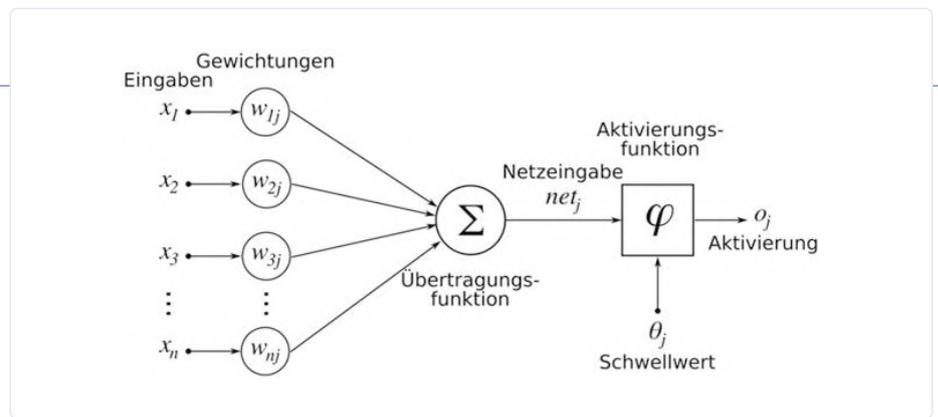


Bild 2. Berechnungen innerhalb eines Knotens (https://commons.wikimedia.org/wiki/Artificial_neural_network).

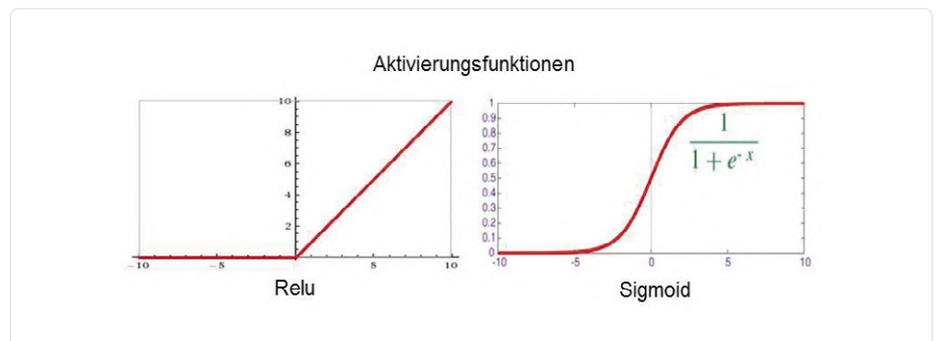


Bild 3. Häufig eingesetzte Aktivierungsfunktionen.

-1 bis +1 gefüllt, das Netz ist also strohdumm und liefert demzufolge auch nur Zufallsresultate. Zum Training werden dem Netz Daten und bekannte Soll-Ausgangswerte zugeführt. Nach der Berechnung mit diesen Werten wird das Ergebnis mit dem Sollwert verglichen und die Differenz von einer Verlustfunktion dokumentiert. Nun folgt der Lernprozess, auch Backpropagation genannt. Beginnend in der letzten Schicht werden die Gewichte in kleinen Schritten so justiert, dass der Verlust minimiert wird. Nach vielen (oft millionenfachen) Trainingsrunden mit unterschiedlichen Eingangsdaten haben alle Gewichte die zur Aufgabenstellung passenden Werte und das NN kann nun neue, ihm unbekannte Daten analysieren.

Eine besonders zur Bild- und Audioverarbeitung geeignete Architektur stellt das *Convolutional Neural Network* (zu Deutsch: Falten des Neuronales Netzwerk) dar, das auch vom Maixduino unterstützt wird. Bei einem solchen CNN sind die Neuronen (zumindest in einigen der Schichten) zweidimensional angeordnet, was zu zweidimensionalen Eingangsdaten wie zum Beispiel einem Bild passt. Gegenüber dem in Bild 1 angedeuteten Netzwerk, bei dem die Aktivität eines Neurons von allen Neuronen der vorhergehenden Schicht (über unterschiedliche Gewichtsfaktoren) abhängt,

ist die Abhängigkeit beim CNN vereinfacht und lokal beschränkt. Die Aktivität eines Neurons hängt hier nur von den Werten von (beispielsweise) 3 x 3 Neuronen ab, die in der Schicht vor ihm angesiedelt sind - die Gewichtsfaktoren sind dabei gleich. Mit einem solchen Netzwerk lassen sich besonders gut kleinräumige Strukturen wie Linien, Rundungen, Punkte und weitere Muster erkennen. In den nachfolgenden Schichten werden dann komplexere Details und schließlich ganze Gesichter detektiert.

Programmtechnisch stellt ein NN ähnlich wie bei einer Tabellenkalkulation eine Anordnung von Zellen mit vorgegebenen Berechnungsanweisungen dar, die bei der Ausführung auf mehrdimensionale Matrizen mit den Gewichten zugreifen. Bei der Klassifikation neuer Eingangsdaten werden alle Schichten vom Eingang bis zum Ausgang durchlaufen und die Ausgänge reflektieren die jeweilige Wahrscheinlichkeit des dem Knoten zugeordneten Resultats. Gut trainierte NNs erreichen dabei durchaus Werte um die 0,9. Beim Training erfolgt nach der Inferenz die Backpropagation mit einem Durchlauf von hinten bis zum Eingang zwecks Justierung der Gewichte.

Klingt im ersten Moment kompliziert, aber es stehen zur Umsetzung zahlreiche Hilfsmittel

und Bibliotheken zur Verfügung, die ich im weiteren Artikelverlauf vorstelle.

Linux und Python

Bereits im ersten Artikel dieser Miniserie habe ich darauf hingewiesen, dass Sie bei der Befassung mit dem Thema KI mit einem Lernaufwand rechnen müssen. Am leichtesten haben Sie es in dieser Welt auf der Linux-Plattform, denn hier stehen die meisten Werkzeuge kostenlos und mit guter Qualität zur Verfügung. Außerdem bietet Linux den gleichen Komfort wie Windows, ist nur anders (meistens besser) verpackt. Meine Präferenz ist Ubuntu, das auch als LTS-Version (Long Term Support) verfügbar ist und mindestens 4 Jahre unterstützt wird. Aber auch die anderen Derivate wie Debian, Mint usw. sind genauso brauchbar, die Wahl ist Geschmackssache. Linux lässt sich neben Windows z.B. in einer virtuellen Maschine installieren, Sie benötigen keinen Extra-Rechner.

Und warum Python? Das ist doch eine interpretierende Programmiersprache, so werden mache sagen, also potentiell langsam. Dieser Nachteil wird durch zahlreiche Vorteile aufgehoben: Zum ersten verzichtet Python auf den Schnickschnack aus Klammern und Semikolons und führt die Blockbildung durch Einrücken der Zeilen durch. Es verfügt über mächtige Datenstrukturen wie Listen, Tupel, Sets und Dictionaries; darüber hinaus ist in Python die Matrizenberechnung bereits voll integriert. Weitere große Vorteile sind die verfügbaren KI-Frameworks und Bibliotheken, die sich aufgrund der hohen Modularität ganz oder teilweise einbinden lassen. Diese sind meistens in C++ geschrieben und weisen daher die erforderliche Performance auf. Und das alles lässt sich mit wenigen Anweisungen bequem installieren.

Wenn Sie loslegen wollen, installieren Sie bitte das von Ihnen favorisierte Linux sowie pip3 und Python 3, Anweisungen dazu sind im Netz reichlich vorhanden.

Maixduino spricht MicroPython

Damit Python auch auf Systemen mit weniger Hauptspeicher laufen kann, steht eine abgespeckte Version unter der Bezeichnung MicroPython zur Verfügung; sie ist auf Plattformen wie Maixduino, ESP32 und anderen installierbar. MicroPython enthält neben einem ausgewogenen Befehls-Repertoire um die 55 Zusatzmodule für zahlreiche mathematische und systemtechnische Funktionen. Um neue Versionen oder KI-Modelle einzupflegen, ist das Flashing-Tool Kflash erforderlich.

Installation von Kflash unter Linux:

- Downloaden der Version 1.5.3 oder höher von Weblink [2] als gepackte Datei `kflash_gui_v1.5.3_linux.tar.xz`.
- Transfer in einen Ordner Ihrer Wahl.
- Entpacken mittels Befehl `tar xvf kflash_gui_v1.5.3_linux.tar.xz`.
- Wechseln in den neu angelegten Ordner `/kflash_gui_v1.5.2_linux/kflash_gui`.
- Starten mit `./kflash_gui` (bei Startproblemen gegebenenfalls in den Eigenschaften dieser Datei das Häkchen bei *Datei als Programm ausführen* setzen).

Damit wird die grafische Oberfläche von Kflash (**Bild 4**) gestartet und es können nun Firmware oder KI-Modelle in den Maixduino geladen werden.

Installation Firmware auf Maixduino

Wenngleich der Maixduino bei der Auslieferung bereits mit MicroPython ausgestattet ist, sollte man stets die neueste Version nachladen. Beim Schreiben dieser Zeilen hat die entsprechende Firmware die Bezeichnung v0.5.0 und diese steht über den Weblink [3] zum Download bereit. Wählen Sie `maixpy_v0.5.0_8_g9c3b97f` oder höher und im nächsten Bild die Variante `maixpy_v0.5.0_8_g9c3b97f_minimum_with_ide_support.bin` oder höher, eine ca. 700 kB große Datei, die auch eine Unterstützung für die MaixPy-IDE enthält.

Mit Hilfe von Kflash ist die neue Firmware schnell installiert. Über den Button *Open File* wird diese ausgewählt und nach der Einstellung von *Board*, *Port*, *Baudrate* und *Speed mode* wie in Bild 5 startet ein Klick auf *Download* den Ladevorgang. Danach kann man bereits mit einem Terminalemulator (z.B. Putty) über den Port `/dev/ttyUSB0` beim Maixduino die ersten Python-Befehle ausführen.

Hier ein kleines Beispiel mit Array-Befehlen:

```
>>> # Python-Prompt
>>> import array as arr # Import Array-Modul
>>> a = arr.array('i',[1,2,3]) # Anlegen Array a mit Integern
>>> b = arr.array('i',[1,1,1]) # Anlegen Array b mit Integern
>>> c = sum(a + b) # Summe aus allen Array-Werten bilden
>>> print(a,b,c) # und ausgeben
array('i', [1, 2, 3]) array('i', [1, 1, 1]) 9 # Ausgabe
>>>
```

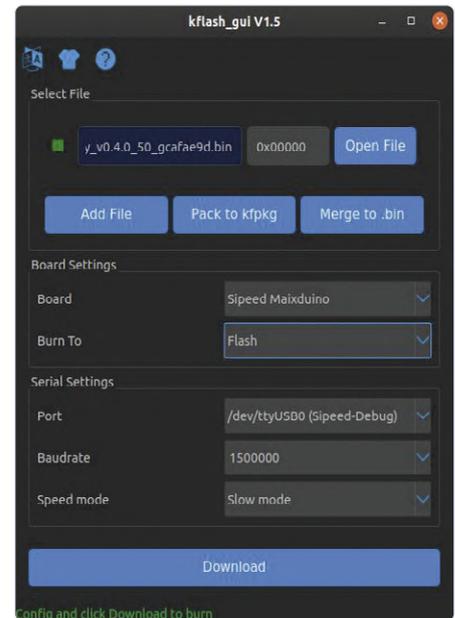


Bild 4. Oberfläche von Kflash.

Mit Bibliotheken wie *numpy* (unter Linux) oder *umatrix* stehen noch mehr Funktionen zur Verfügung.

Installation MaixPy-IDE

Viel komfortabler gehen Entwicklung und Tests mit der Entwicklungsumgebung MaixPy-IDE vonstatten. Darüber können Python-Programme entwickelt und getestet sowie auf den Maixduino geladen und ausgeführt werden. Zusätzlich stehen, wie in **Bild 5** dargestellt, Tools zur Bildanalyse zur Verfügung. Die Installation wird wie folgt durchgeführt:

- Download der Version `maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z` oder höher von Weblink [4].
- Transfer in einen Ordner Ihrer Wahl.
- Entpacken mittels Befehl `tar maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z`.
- In den neuen Ordner `maixpy-ide-linux-x86_64-0.2.4-installer-archive` wechseln

und folgende Befehle eingeben:

```
./setup.sh  
./bin/maixpyide.sh
```

Danach startet die IDE. Bei nachfolgenden Starts ist natürlich nur der letztgenannte Befehl erforderlich.

Damit stehen sämtliche Werkzeuge zur Implementierung von KI-Modellen zur Verfügung. Nachfolgend wird eine Gesichtserkennung getestet.

Gesicht erkannt!

Zur Gesichtserkennung wird ein bereits trainiertes KI-Modell eingesetzt, das bereits mehrere Tausend Gesichter nach charakteristischen Merkmalen analysiert hat - die Gewichte des eingesetzten NN sind entsprechend justiert. Das Modell steht unter dem Weblink [5] unter der Bezeichnung *face_model_at_0x300000.kfpkg* zum Download bereit. Basis der Entwicklung ist hierbei das KI-Framework Yolo2 (You Only Look Once), das die Bildobjekte in mehrere Zonen unterteilt, diese gesondert analysiert und dadurch hohe Erkennungsraten erreicht (auf die KI-Frameworks werde ich im weiteren Verlauf der Serie noch eingehen). Das KI-Modell für den KI-Prozessor ist im *kfpkg*-Format gepackt und muss im Maixduino an die Adresse 0x300000 geflasht werden. Dieses lässt sich ebenfalls mit Kflash erledigen; hierfür einfach die Datei mittels *Open File* aufsuchen und anschließend mit den Parametern von Bild 4 auf das Board laden.

Zur komfortablen Abwicklung des Python-Skripts kommt die MaixPy-IDE zum Einsatz, mit der man Programme entwickeln, testen und in den Maixduino transferieren kann. Bild 5 zeigt die in drei Fenster eingeteilte Oberfläche:

- **Editor**, links-oben: In diesem Bereich erfolgt die Programmeingabe mit Syntax-Highlighting.
- **Terminal**, darunter: Darstellung der Programmausgaben.
- **Bildanalyse**, rechts: Hier erfolgen die Darstellung von Bildern und ihre spektrale Aufteilung in die Farben Rot, Grün, Blau.

Neben anderen verfügbaren Buttons und Menüpunkten sind die beiden Knöpfe links unten wichtig: Mit der „Büroklammer“ wird die Verbindung zum Maixduino über den Port „ttyUSB0“ hergestellt (Farbe Grün) bzw. getrennt (Rot). Ein grünes Dreieck darunter startet das Skript; danach wandelt sich dieser Button in einen roten Punkt mit „x“ und dient zum Stoppen des Programms.

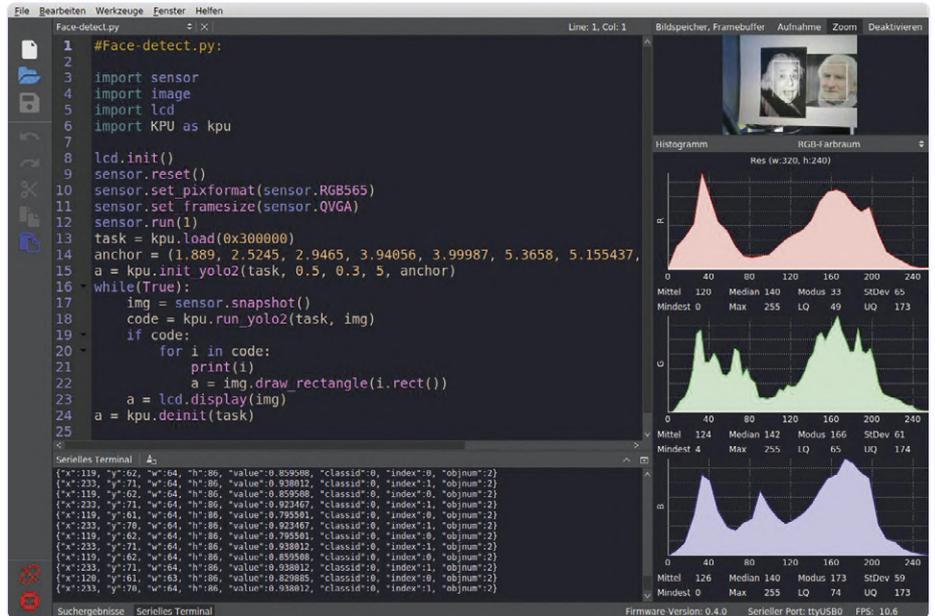


Bild 5. Oberfläche der MaixPy-IDE.

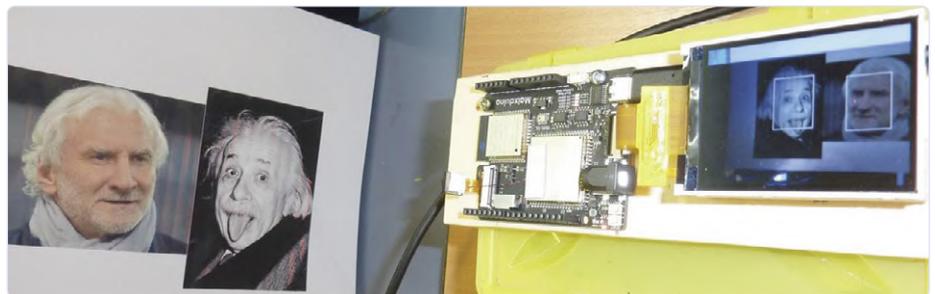


Bild 6. Testaufbau für die Gesichtserkennung.

Zur Durchführung des Tests habe ich die Gesichter von zwei öffentlich bekannten Personen (Albert Einstein und Rudi Völler) ausgedruckt und an eine Wand gepinnt. Die Auswahl war rein zufällig, aber man sagt beiden Gesichtern eine gewisse Ähnlichkeit nach, was von mir allerdings nicht bestätigt wird. Bei der Aufnahme dieser Bilder wurden die Gesichter sofort erkannt und mit einem Rahmen markiert. Es ist darauf zu achten, dass die Darstellung wie gezeigt im Querformat erfolgt, ansonsten sinkt die Erkennungsrate spürbar ab.

Das Programm *Face-detect.py* finden Sie im Download-Ordner auf der Elektor-Webseite [6]. Seine Kürze reflektiert wiederum die Leistung der eingesetzten Bibliotheken. Zum Start erfolgen die Einbindung der benötigten Bibliotheken für Kamera, LCD und KPU sowie deren Initialisierung. Danach wird das NN ab Adresse 0x300000 in die KPU geladen. Bei

der Initialisierung des KI-Modells durch den Befehl `kpu.init_yolo2` werden zusätzliche Konstanten zur Einstellung der Genauigkeit und Optimierung übergeben. Nun wird in einer endlosen `while`-Schleife die Bildklassifikation durchgeführt, ein Bild aufgenommen und dem NN zugeführt. Sind Gesichter entdeckt worden, erhält die Variable `i` für jedes Gesicht die Koordinaten und Größe eines Markierungsrahmens, der anschließend in das Bild eingezeichnet wird. Abschließend erfolgen die Ausgabe des Bildes (auf dem LCD-Panel) und der Markierungsdaten (auf dem seriellen Terminal). Weitere Details zu den KPU-Befehlen sind unter dem Link [7] abrufbar.

In der MaixPy-IDE wird das Bild rechts ebenfalls gezeigt und darunter das dazugehörige Farbspektrum dargestellt. Falls Sie diese Information nicht benötigen, ist das rechte Bildfenster mit dem *Deaktivieren*-But-



Bild 7. Darstellung auf dem Maixduino-LCD.

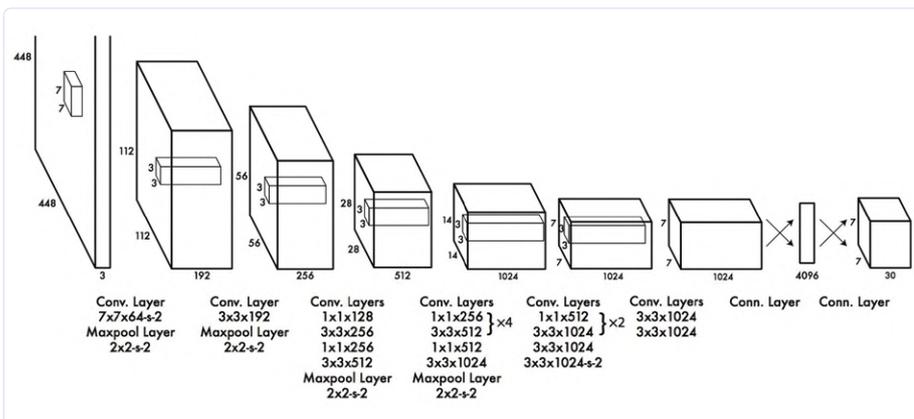


Bild 8. Netzarchitektur zur Gesichtserkennung (Quelle: <https://bit.ly/3cK3DUR>).

ton abschaltbar.
Zur besseren Handhabbarkeit habe ich Maixduino und LCD auf ein kleines Brett-

chen montiert und die Kamera nach vorne ausgerichtet (siehe **Bild 6**). Damit lassen sich leicht echte Gesichter, gedruckte Bilder oder

Bildschirminhalte erfassen und analysieren. Die Darstellung auf dem LCD-Panel sieht man in **Bild 7**, Ähnlichkeiten der Personen sind nicht zu erkennen.

Aber was steckt hinter diesem Yolo2-Modell? Das Neuronale Netz verfügt über 24 Convolutional-Schichten und zwei voll verbundene Ausgangs-Layer (siehe **Bild 8**). Dazwischen sind einige Maxpool-Layer als Filter zwischengeschaltet, um Komplexität herauszunehmen und den Hang zum „Auswendiglernen“ zu verringern. Es fällt auf, dass weitgehend eine Fenstergröße von 3x3 zur Detailerkennung genutzt wird. Exakt diese wird von der KPU-Hardware unterstützt, was dem Maixduino für solche Aufgaben eine hohe Leistungsfähigkeit sichert.

Andere bekannte NN-Strukturen verfügen sogar über mehrere Hundert Schichten, haben Rückführungen oder andere Extras. Der Kreativität sind in diesem Bereich keine Grenzen gesetzt, vieles hängt vom Budget, sprich Rechenleistung, ab.

Und weiter geht's!

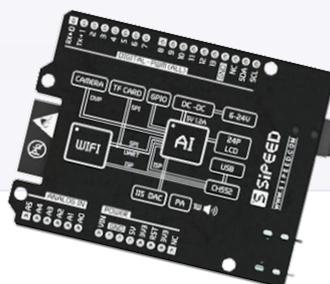
Die leistungsstarke Hardware und die bereits jetzt verfügbare Software-Umgebung zeigen, dass der Maixduino für den Einstieg in die Künstliche Intelligenz gut geeignet ist. Er bietet sich aufgrund des geringen Stromverbrauchs gut für den Einsatz in mobilen Geräten mit bereits trainierten Neuronalen Netzen an. In einem dritten Teil der Serie werde ich Ihnen zeigen, wie man ein eigenes Neuronales Netz entwickeln, trainieren und ausführen kann. Die Schnittstelle zum Entwickler bildet dabei das KI-Framework Keras, das auch als komfortabler „Legobaukasten für KI“ bezeichnet wird. Außerdem erhalten Sie Hinweise, wie man den ESP32 auf dem Board programmieren kann - zum Beispiel um analoge Werte zu erfassen. Bleiben Sie neugierig!

200023-B-01



IM ELEKTOR-STORE

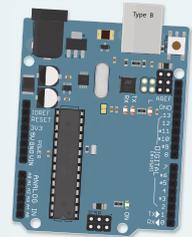
> **Sipeed MAix BiT Kit for RISC-V AI+IoT**
www.elektor.de/sipeed-maix-bit-kit-for-risc-v-ai-iot



WEBLINKS

- [1] **KI für Einsteiger (1), Elektor Mai/Juni 2020:** www.elektormagazine.de/200023-01
- [2] **Kflash:** https://github.com/sipeed/kflash_gui/releases
- [3] **Maixduino-Firmware:** <http://dl.sipeed.com/MAIX/MaixPy/release/master/>
- [4] **MaixPy-IDE:** http://dl.sipeed.com/MAIX/MaixPy/ide/_/v0.2.4/maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z
- [5] **KI-Modelle:** <http://dl.sipeed.com/MAIX/MaixPy/model>
- [6] **Software:** www.elektormagazine.de/200023-B-01
- [7] **KPU-Befehle:** <https://maixpy.sipeed.com/en/libs/Maix/kpu.html>

Interessiert an Arduino?



Elektor Special Einstieg in die Elektronik mit Arduino

Dieses Buch bietet Ihnen einen geordneten Einstieg in die Elektronik mit dem Arduino Board. Es bietet die ideale Plattform für Einsteiger, die sich auch einmal an die Programmierung und Entwicklung mit einem Mikrocontroller wagen wollen. Und damit der Einstieg schnell und einfach gelingt, beschränkt sich dieses Arduino-Special auf das Wesentliche.

Für die beschriebenen Praxisprojekte braucht man gar nicht viel Material. Die Schaltungen werden auf einem kleinen Experimentier-Steckboard aufgebaut und gegebenenfalls mit einem Arduino Uno verbunden, den man an einem Windows-PC programmiert. Zu allen Schaltungen gibt es übersichtliche Aufbaubeispiele.

22
Projekte

- 01 Strom & Spannung: Wie alles begann
- 02 Arduino-Hardware
- 03 Arduino-Programmierung
- 04 Der Stromkreis
- 05 Mit dem Multimeter messen
- 06 Schaltpläne und Experimentierboards
- 07 Schaltpläne erstellen
- 08 Breadboardansichten mit Fritzing
- 09 Schaltungen online simulieren
- 10 Unverzichtbar: Widerstände (Teil 1)
- 11 Praxis mit Widerständen (Teil 2)
- 12 Veränderliche Widerstände
- 13 Dioden: Einbahnstraße für den Strom
- 14 Der Transistor-Schalter
- 15 Elektromagnetismus
- 16 Relais und Motoren
- 17 OpAmps: Operationsverstärker
- 18 Kondensatoren
- 19 Timer NE555
- 20 PWM und analoge Werte mit Arduino
- 21 7-Segment Temperaturanzeige
- 22 Einführung ins Löten und LCDs

Mehr Infos & bestellen:
www.elektor.de/19286



Gewinnen Sie
75.000€

Marketing-Paket von
Elektor und einen Stand
auf der **electronica**

COMPETE TO
LAUNCH
YOUR STARTUP AT
ELECTRONICA
2020

Ist Ihr Elektronik-Start-up bereit die
Lösungen Ihres Unternehmens auf dem
globalen Markt zu präsentieren?

 **DANN BEWERBEN SIE SICH JETZT!**

Besuchen Sie elektormagazine.de/fastforward, um die
Regeln, Bedingungen und Konditionen zu lesen und Ihre
Bewerbung einzureichen. Viel Erfolg!

VORTEILE DER FINALISTEN:

- **Erster Preis = 75.000 €** Marketing-Paket von Elektor, plus
einen Stand auf der **electronica 2022**
- **Zweiter Preis = 50.000 €** Marketing-Paket von Elektor
- **Dritter Preis = 25.000 €** Marketing-Paket von Elektor
- Erhalten Sie einen **exklusiven, besonders preisgünstigen**
(nur 1.905 €!) Stand (5 m²) auf der **electronica 2020**
- Erhalten Sie **große Aufmerksamkeit** vor, während und nach
der **electronica 2020**. (10.-13.11.2020)



Sign up
here